Depth-Guided Privacy-Preserving Visual Localization Using 3D Sphere Clouds

Heejoon Moon[1]
wilko97@hanyang.ac.kr
Jongwoo Lee[1]
sanngu5@hanyang.ac.kr
Jeonggon Kim[2]
drgon22@hanyang.ac.kr
Je Hyeong Hong*[1,2]
jhh37@hanyang.ac.kr

[1] Department of Artificial Intelligence
Hanyang Univeristy
Seoul, Republic of Korea

[2] Department of Electronic Engineering
Hanyang Univeristy
Seoul, Republic of Korea

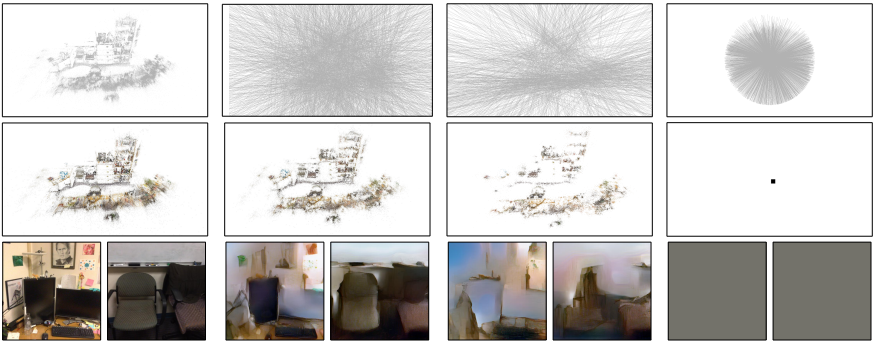
Abstract

The emergence of deep neural networks capable of revealing high-fidelity scene details from sparse 3D point clouds has raised significant privacy concerns in visual localization involving private maps. Lifting map points to randomly oriented 3D lines is a well-known approach for obstructing undesired recovery of the scene images, but these lines are vulnerable to a density-based attack that can recover the point cloud geometry by observing the neighborhood statistics of lines. With the aim of nullifying this attack, we present a new privacy-preserving scene representation called *sphere cloud*, which is constructed by lifting all points to 3D lines crossing the centroid of the map, resembling points on the unit sphere. Since lines are most dense at the map centroid, the sphere cloud mislead the density-based attack algorithm to incorrectly yield points at the centroid, effectively neutralizing the attack. Nevertheless, this advantage comes at the cost of i) a new type of attack that may directly recover images from this cloud representation and ii) unresolved translation scale for camera pose estimation. To address these issues, we introduce a simple yet effective cloud construction strategy to thwart new attack and propose an efficient localization framework to guide the translation scale by utilizing absolute depth maps acquired from on-device time-of-flight (ToF) sensors. Experimental results on public RGB-D datasets demonstrate sphere cloud achieves competitive privacy-preserving ability and localization runtime while not excessively compensating the pose estimation accuracy compared to other depth-guided localization methods.

1 Introduction

Visual localization, which refers to the task of estimating the 6-DOF camera pose from an input image, is a key computation in autonomous driving, extended reality (XR) [1, 4] and robotics [4, 18, 22]. While a full taxonomy of localization algorithms exists in the literature, the mainstream pipeline to this date comprises the following steps: i) build a sparse 3D point

* denotes corresponding author



(a) Ground Truth (b) ULC [35] (c) PPL [16] (d) **Sphere cloud (ours)**

Figure 1: (*Top*) visualization of different 3D scene representations (*Office1 manolis* from 12 Scenes [38]). ULC [35] denotes uniform line cloud and PPL [16] denotes paired-point lifting. (*Middle*) recovered 3D points from the geometry revealing attack [6]. (*Bottom*) images reconstructed via InvSfM [28] using the recovered 3D points. Since our sphere cloud always results in points recovered at the sphere centre, the recovered scene images are blank.

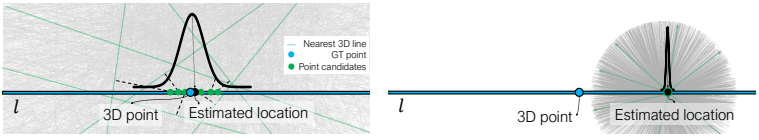
cloud of the scene via structure-from-motion (SfM) [31], ii) match keypoints of the query image against the features in the point map and iii) perform perspective-*n*-points to obtain the camera pose. The point cloud and descriptors are either stored on the server for cloud-based localization or distributed to the client (*e.g.* a robot or XR device) for real-time localization.

Until recently, it was perceived that these point maps, which may often comprise a private or confidential area/objects, are usually sparse enough to discourage any attempt by curious intruders or malicious clients to reveal scene details from the 3D points. Nevertheless, the work of Pittaluga *et al.* [28] called InvSfM showed possibility of recovering high-fidelity scene images from the sparse point cloud, raising significant privacy concerns when using the barebone point maps for localization. Currently, one of the most widely known approaches to mitigating this issue is to conceal the point map as a *line cloud*, which is constructed by lifting each point to a 3D line [16, 35], subsequently hiding the point locations and disabling direct image synthesis using InvSfM. Unfortunately, this line of works is potentially vulnerable to the density-based attack [6] (see Fig. 1 for an example), which can effectively reverse the 3D lines back to points using the neighbourhood statistics of the lines. Providing a full defense against such attack is yet an unaccomplished goal and serves as our main motivation.

In this work, we present a new privacy-preserving scene representation called *sphere cloud* in an effort to nullify aforementioned geometry-revealing attack [6]. The sphere cloud, which is simply constructed by lifting points to 3D lines passing through the centroid of the point cloud (which can be viewed as points on the unit sphere centered at the map centroid), has the advantage of completely disabling the geometry-revealing attack [6] by forcing the neighbourhood line statistics to lead to a degenerate point recovery (see Sec. 3 for details). Unfortunately, employing a sphere cloud for privacy-preserving visual localization is not straightforward due to two issues, that i) a new type of attack (discussed in Sec. 3) may partly reveal scene details about the map centroid and ii) the camera pose can only be retrieved up to unknown scale. We tackle the first issue by proposing a simple effective strategy to hinder new attack and address the second issue by utilizing calibrated depth maps that can be easily acquired from an on-device time-of-flight (TOF) sensor to resolve the translation scale.

Our contributions in this work are summarized as follows:

- a novel privacy-preserving scene representation called *sphere cloud* which completely avoids known density-based attack and disables recovery of the point cloud geometry,



(a) Uniform line cloud (ULC) [35] (b) Sphere cloud (ours)

Figure 2: A motivating illustration for the sphere cloud. As shown in (a), the density-based geometry-revealing attack in [5] recovers the point location of each line by constructing a histogram of point candidates on the line that are closest to K-nearest neighbouring lines and finding the peak of the histogram. While this method often yields good point estimates for uniform line clouds, all lines from the sphere cloud in (b) intersect at the map centroid, and consequently the points estimated via peak finding are incorrectly recovered at the centroid.

- a simple yet effective strategy based on cloud sparsification and descriptor augmentation to thwart a new type of attack from breaching the sphere cloud, and
- to the best of our knowledge, the first privacy-preserving framework to leverage raw depth information from a ToF sensor for efficient camera pose estimation.

2 Related work

Revealing private scene details from sparse point cloud The first method that succeeded in revealing high-fidelity scene details from a sparse point cloud was proposed by Pittaluga *et al*. [28], in which a network called InvSfM based on cascaded U-Net [30] is employed to reconstruct a scene image from a set of inputs including 2D locations of the projected 3D points as well as corresponding depths, RGB values and SIFT descriptors. As noted in [35], this raised alarms as any confidential maps (*e.g*. inside a factory) or public maps with temporary private objects inadvertently obtained by a user can now be revealed in detail. While extensions of this work have been proposed to reconstruct images without keypoint descriptors [32] or with different types of descriptors [8], the pretrained InvSfM model is still widely used as the baseline for analyzing the privacy-preserving capability [5, 16, 20, 24, 26].

Privacy-preserving 3D scene representations With the aim of obstructing use of InvSfM for scene image reconstruction, Speciale *et al*. [35] proposed line cloud in which each point is represented as a randomly oriented 3D line passing through the original point, intending to conceal the scene geometry by introducing ambiguities in the point locations. While this was initially perceived as an effective strategy to block attempts for revealing scene details and extended to simultaneous localization and mapping (SLAM) [32], it was later shown by Chelani *et al*. [5] that line clouds with uniformly distributed line directions are vulnerable to a density-based geometry-inversion attack that can accurately recover the scene points (more details at the end of Sec. 2), from which the scene images can subsequently be revealed (see Fig. 1 (b)). While this weakness was addressed in [16] by drawing 3D lines through random pairs of 3D points to induce combinatorial complexity in point cloud recovery, it is not fully impervious to the geometry-revealing attack [5] as observed in the second row of Fig. 1 (c). The most similar work to our approach is [20], in which all 3D lines intersect through one of two pre-defined 3D locations to reduce the effectiveness of the density-based attack [5]. However, this method does not theoretically guarantee full defense against [5] and moreover, it can be vulnerable to another type of attack involving direct image synthesis at the intersections (see Sec. 3.1).

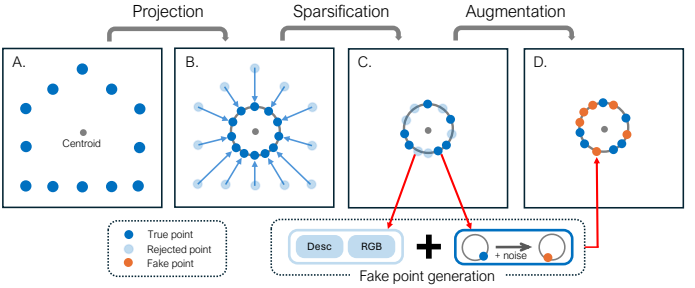
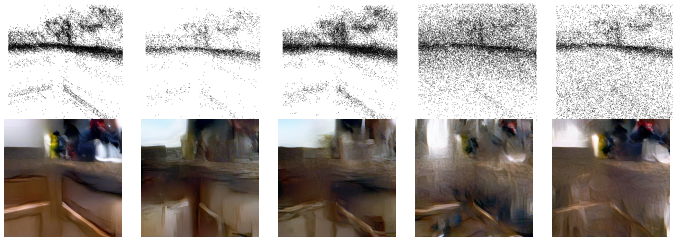


Figure 3: An overview of our complete strategy for constructing a sphere cloud. (A) we find the centroid of the sparse 3D point map. (B) we create a basic sphere cloud by projecting the 3D points onto the unit sphere centred at the map centroid. (C) we discard a portion of sphere points from the sphere cloud but keep their RGB values and SIFT descriptors. (D) we generate fake points around the remaining points with their RGB values and SIFT descriptors recycled from the rejected points. Since the basic construction from (A)+(B) may be prone to a new attack based on direct image synthesis, we enhance the strategy through (C)+(D).

Other types of scene representations include the work of Geppert *et al.* [11], in which the sparse point cloud is divided into three 1D partial maps stored in separate servers for enhanced security at the cost of reduced off localization accuracy and runtime. Pan *et al.* [24] proposed to pair up 3D points and permute coordinates between each pair of points to disallow meaningful reconstruction of the scene while enabling accurate localization, but the permutation process incurs combinatorial search over the correct camera pose, drastically slowing the localization speed. Overall, these approaches are not susceptible to the geometry-inversion attack [5] but they are computationally much more involved than line cloud-based approaches, impeding their practical use for efficient real-time localization. Currently, no representation can fully bypass above attack while maintaining real-time localization speed.

Localization using 3D line clouds The classic absolute camera pose estimation problem involving a 3D point cloud can be solved with an efficient perspective-*n*-point (pnP) solver [9, 14, 25] derived from the 2D-3D point-to-point constraints. In contrast, line clouds can only introduce weaker constraints between 2D points and 3D lines. Speciale *et al.* [35] noted absolute pose estimation with line clouds is identical to the problem of generalized relative pose estimation, and proposed a perspective-6-lines (p6L) algorithm based on the minimal solver for generalized relative pose estimation [36]. Due to the intrinsic flexibility of generalized cameras, p6L yields 64 pose candidates for each of six 2D point-3D line correspondences from which the correct solution needs to be identified via geometric verification. Hence, employing p6L contributes to much increased runtime when compared with p3P that only yields 4 pose candidates for each of three 2D-3D point correspondences.

Geometry-revealing density-based attack for line clouds Chelani *et al.* [5] proposed an algorithm for recovering the original points from a uniform 3D line cloud [35]. This work is motivated by the empirical observation that for any two distinct 3D points and their lifted 3D lines, the points on the lines which are closest to the counterpart lines are likely to be in the proximity of the original 3D points. As shown in Fig. 2(a), this result is extended to consider the closest points to multiple neighbouring lines as point candidates (green) for each line. The final point location is estimated by finding the peak (black) of the histogram of these point candidates which is usually close to ground truth (blue) as long as the line directions are uniformly distributed. We neutralize this attack by essentially breaking this assumption as will be described in Sec. 3, leading to an incorrect recovery (see Fig. 2(b)).



(a) Pseudo-GT (b) $\sigma^2 = 0$ (c) $\sigma^2 = 0.01$ (d) $\sigma^2 = 0.1$ (e) $\sigma^2 = 1$

Figure 4: Effect of our fake point generation strategy on the direct image-synthesis attack. We use the $\eta = 0.33$ setting whereby 67% sphere points are discarded and replaced by fake points recycling the SIFT descriptors of the rejected points. Pseudo-GT stands for an image reconstructed via InvSfM [28] about the sphere centre using the original points. σ denotes the standard deviation of Gaussian noise injected to generate fake points (see Sec. 3.2). We determine $\sigma^2 = 0.1$ as the "sweet" spot as it hides both the scene geometry and image details.

3 Sphere cloud

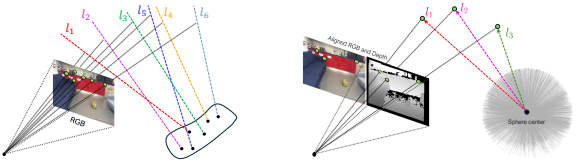
Motivations As shown in Fig. 2(a), [5] recovers the 3D points by constructing a histogram of point candidates for each 3D line (i.e. a set of points on the line each of which is closest to one of the neighboring lines) and finding the peak of this histogram. Now, if all lines are lifted to meet at a single point $\mathbf{c} \in \mathbb{R}^3$, then the point candidates for each line will always be located at the intersection point as any two lines are the closest at $\mathbf{c}$. Consequently, the peak of the histogram is always at $\mathbf{c}$, leading to a degenerate recovery and thereby voiding the attack (see Fig. 2(b)). This motivates us to have all lines lifted to meet at a single point.

Unfortunately, the above representation is not sufficient to yield a unique camera rotation. Since the 3D lines intersecting at a single point can be viewed as rays from a virtual camera centered at the intersection point (see Fig. 5(b)), estimating camera pose from these lines resembles the relative pose estimation problem between the query camera and virtual camera (also noted in [5]). Out of 4 possible configurations [12] between the two cameras, we can choose the correct solution only if the cheirality is enforced on the lifted 3D lines (more details in [21]). This serves as motivation for storing each line $\mathbf{l}_i$ as a point $\hat{\mathbf{x}}_i \in S^2$ on the unit sphere centered at $\mathbf{c}$ such that the original point is always along the positive direction of $\hat{\mathbf{x}}_i$.

3.1 Basic construction procedure and limitations

Constructing a basic 3D sphere cloud involves two straightforward steps (steps A and B in Fig. 3). First, we set the intersection point as the mean centroid of the 3D point cloud to ensure the resulting line directions are roughly evenly distributed for stable localization (see [21] for discussions). Second, we project all 3D points onto the unit sphere centred at the map centroid to create a basic 3D sphere cloud. Unfortunately, there are two major issues with deploying this basic implementation for privacy-preserving visual localization.

Possible attack based on direct image synthesis While the sphere cloud does not leak any scene geometry, an intruder may seek to directly reveal images from the sphere cloud. The simplest approach is to project the sphere points to a virtual image plane and feed the projected points and their descriptors to InvSfM [27]. Although the intruder is confined to viewpoints about the map centroid, Fig. 4(a) shows this attack can partly reveal the scene. We aim to thwart this attack through an enhanced construction strategy in Sec. 3.2.



(a) p6L solver for line clouds [65] (b) p3P solver for the sphere cloud

Figure 5: Comparison of minimal solvers for privacy-preserving localization. In (b), the depth map allows lifting keypoints to 3D and estimating pose using an efficient p3P solver.

Unresolved translation scale As mentioned earlier in Sec. 3 camera pose estimation using the sphere cloud boils down to the perspective relative pose estimation problem, meaning the translation scale is unknown [23]. As many modern commercial devices such as iPad or HoloLens 2 comprise depth sensors, we attempt to efficiently leverage calibrated raw depth maps from the on-device time-of-flight (ToF) sensor to retrieve absolute scale (see Sec. 3.3).

3.2 Enhanced construction strategy

To hinder direct image synthesis from the sphere points, we add fake points to the sphere cloud. This is inspired by the observation that embedding fake points between real keypoints degrades the quality of scene images reconstructed via InvSfM (see Fig. 4).

Cloud sparsification As excess number of fake points can make the sphere dense and slow the localization speed, we avoid this by keeping the total number of sphere points fixed. If the desired proportion of true positive sphere points is η, then we discard $1-\eta$ of all points.

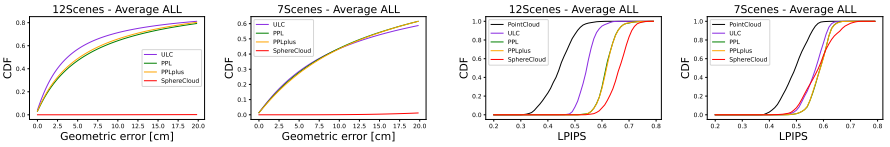
Generating fake point locations We employ a simple approach of adding Gaussian noise to the coordinates of existing sphere points, i.e. where $\hat{\mathbf{x}}_i \in S^2$ is the i-th sphere point, $\mathbf{z}_{ij}$ is the j-th fake point generated in the proximity of $\hat{\mathbf{x}}_i$ and $\varepsilon \sim N(\mathbf{0}, \sigma^2 \mathtt{I})$ is Gaussian noise with σ^2 set to 0.1. The number of fake points generated per sphere point is constant. Since the total number of sphere points remains constant, we generate $(1-\eta)/\eta$ fake points for each remaining sphere point (*e.g.* if $\eta = 33\%$, then we discard 67% points and create two new fake points per remaining sphere point.)

Assigning fake point descriptors via recycling After assigning the fake point locations, we need to designate a realistic feature descriptor to each fake point. We refrain from using keypoint descriptors extracted from a large database of images [10] as there is a potential risk of this database being hijacked in which case the fake points can be easily pruned. We also do not adopt a learning-based scheme as the generated descriptors may potentially be detected by training a discriminator network. Instead, we resort to a simple strategy of recycling the descriptors of discarded sphere points. Since the number of fake points is equal to the number of rejected points, this amounts to a simple permutation of descriptors from the rejected points followed by assignment of these features to the fake point locations.

As shown in Fig. 8, this strategy effectively mitigates the issue of direct image synthesis for the sphere cloud. Additionally, adjusting η controls the trade-off between localization accuracy and privacy-preserving ability as shown in Fig. 8 and Table 2.

3.3 Camera pose estimation using RGB image and depth map

We illustrate a framework for absolute pose estimation using the sphere cloud assuming the query has an aligned pair of RGB image and absolute depth map with known intrinsic.



(a) 3D point reconstruction errors (b) LPIPS [39] of recovered images

Figure 6: Cumulative distributions of (a) the geometric error (e_g) of 3D points recovered using [5] and (b) LPIPS of reconstructed images from InvSfM [28].

Efficient initial pose estimation via perspective-n-points Fusing the query RGB image with its aligned depth map allows us to lift each 2D keypoint $\mathbf{u}_i \in \mathbb{R}^2$ to the 3D space as $\mathbf{p}_i = z_i^{TOF} \mathtt{K}^{-1}[\mathbf{u}_i^\top, 1]^\top$, where $z_i^{TOF} \in \mathbb{R}$ is the depth of $\mathbf{u}_i$ obtained from the ToF sensor and $\mathtt{K} \in \mathbb{R}^{3\times3}$ is the camera intrinsics. We note aligning these 3D keypoints $\{\mathbf{p}_i\}$ with the matched 3D lines $\{\mathbf{l}_i\}$ from the sphere cloud is nearly identical to the perspective-n-points problem except that the 3D points are on the query side and not on the map side (see Fig. 5(b)).

Ideally, the 3D keypoint $\mathbf{p}_i$ should lie along the positive direction of the vector shooting out from the sphere centre and passing through the sphere point $\hat{\mathbf{x}}_i$, i.e. where $[\mathtt{R}|\mathbf{t}]$ defines the query-to-world (sphere cloud) transformation. This geometric constraint can be efficiently solved using a combination of LO-RANSAC [15, 19] and the p3P solver [25], which only needs 4 correspondences compared to 6 required in the absence of depth maps [16, 24, 35]. The final (world-to-query) pose is obtained as $[\mathtt{R}^\top | -\mathtt{R}^\top \mathbf{t}]$, and heavy outliers are pruned by checking the epipolar distance in Eq. (1) and the depth error in Eq. (2). The threshold values used in our implementation can be found in supplementary material.

Pose refinement with depth regularization After an initial pose is obtained, we refine the pose via nonlinear optimization [15]. Since the sphere cloud can be viewed as a special type of line cloud, we follow the direction of other line clouds [16, 35] and partly minimize the square of epipolar distance between the projection of 3D lines derived from the sphere cloud and the 2D query keypoints. The resulting loss function for the i-th keypoint, L_i^e, is

$$L_i^e = \frac{([\mathbf{u}_i^\top, 1]\mathtt{K}^{-\top}\,\mathtt{E}\,\tilde{\mathbf{x}}_i)^2}{(\mathbf{e}_1^\top \tilde{\mathbf{x}}_i)^2 + (\mathbf{e}_2^\top \tilde{\mathbf{x}}_i)^2} \tag{1}$$

where $\mathtt{E} := [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]^\top$ denotes the essential matrix between the sphere cloud and the query camera, and $\tilde{\mathbf{x}}_i = \hat{\mathbf{x}}_i / |\hat{x}_{iz}|$ is the z-normalization of the sphere point $\hat{\mathbf{x}}_i$.

Since (1) is oblivious to the translation scale, we additionally employ a depth regularization term to guide the camera pose to the correct scale. For this purpose, we define another loss

$$L_i^d = (\beta_i - 1)^2, \tag{2}$$

where $\beta_i = z_i / z_i^{TOF}$ is the proportional difference between the predicted depth $z_i(\mathtt{R}, \mathbf{t})$ from the current pose and the sphere cloud and the observed depth z_i^{TOF} (an analytic derivation of $z_i(\mathtt{R}, \mathbf{t})$ can be found in [21]).

The overall cost function iteratively minimized over $\mathtt{R}$ and $\mathbf{t}$ is

$$L = \sum_{i \in \Omega} \left(L_i^e + \lambda\, L_i^d \right), \tag{3}$$

where λ is the hyperparameter empirically set to 10^{-4} and Ω represents all of the 2D–3D correspondences on the query image. (more details in [21]).

4 Experiments

Datasets We used two public RGB-D camera re-localization datasets as presented in [33, 38]. 7-Scenes [33] and 12-Scenes [38] consist of several RGB and depth frames of indoor

(a) Ground truth (b) Point cloud (c) ULC [63] (d) PPL [16] (e) PPL+ [16] (f) Sphere cloud

Figure 7: Images revealed from some test camera poses across different scene representation via InvSfM [28]. (Top) *Apt2 kitchen* in 12-Scenes [38]. (Bottom) *Office* in 7-Scenes [33].

Table 1: Quantitative analysis of the direct image-synthesis attack (see Sec. 3.1) on sphere clouds. Each image is reconstructed from an arbitrary viewpoint at the sphere centre. Since no ground truth images are available for these viewpoints, the metrics are calculated using the images reconstructed from 3D point clouds as pseudo-ground truth, but these are often very noisy for the 7-Scenes dataset as shown in Fig. 8. For the sphere cloud, results are reported across different proportions of true positive sphere points (η).

Dataset	Metric	ULC [63]	PPL [16]	PPL+ [16]	Sphere (η=25%)	Sphere (η=33%)	Sphere (η=50%)
12-Scenes [33]	PSNR (↓)	16.06	11.99	**11.30**	12.70	13.53	14.94
	LPIPS (↑)	0.456	0.539	0.542	**0.568**	0.534	0.488
	SSIM (↓)	0.519	0.440	0.436	**0.372**	0.429	0.493
	MAE (↑)	31.82	55.91	**56.88**	47.19	42.53	35.39
7-Scenes [33]	PSNR (↓)	13.11	11.04	**10.84**	13.41	14.01	15.29
	LPIPS (↑)	0.548	0.602	**0.603**	0.550	0.533	0.498
	SSIM (↓)	0.417	0.390	**0.380**	0.393	0.417	0.471
	MAE (↑)	44.85	59.37	**60.50**	43.25	40.28	34.13

spaces captured with multiple sequences. For 7-Scenes [33], we followed additional procedures in [2] to align depth maps to RGB images (not required for 12-Scenes).

Implementation details We implemented our RGB-D localization pipeline for sphere clouds using the PoseLib library [15] and brought the inversion pipeline from [16].
For our localization experiment, we used the official RGB-D benchmark released by [3] based on the above two datasets [33, 38] which contains sparse 3D point clouds reconstructed using COLMAP [31] and the lists of test images. However, as [3] does not provide the SIFT descriptors [17] required for image recovery, we carefully reconstructed these point clouds ourselves using COLMAP following the same protocol of [3] and used them along with the same set of test images for comparing the privacy-preserving capability of different methods. All our experiments were carried out on a PC with Intel CPU i9-13900K running at 3.0 GHz and a single NVIDIA RTX 4090 graphics card.

Evaluation metrics For the quantitative evaluation of 3D point recovery using [5], we reported the 3D point error $e_g = \|\mathbf{g} - \mathbf{g}^*\|_2$ between the estimated point $\mathbf{g} \in \mathbb{R}^3$ and the original point $\mathbf{g}^* \in \mathbb{R}^3$. For comparing the image reconstruction quality, we used the peak signal-to-noise ratio (PSNR), learned perceptual image patch similarity (LPIPS), structural similarity index measure (SSIM) and mean absolute error (MAE) metrics. For evaluating the

(a) Pseudo-GT (b) ULC [35] (c) PPL [16] (d) PPL+ [16] (e) Ours (25%)(f) Ours (33%)(g) Ours (50%)

Figure 8: Visualization of images directly reconstructed from sphere clouds about the sphere centre. (Top) *apt2 kitchen* in 12-Scenes [33]. (Bottom) *office* in 7-Scenes [33]. (a) is the result of applying InvSfM [28] to the original point cloud. (e), (f), and (g) are the results of the sphere cloud across different proportions of true positive sphere points (η). Note the viewpoints are deliberately chosen to be close to the test poses in Fig. 7 for better comparison.

localization performance, we followed [3] and reported the rotation error as $\Delta\mathtt{R} = \angle(\mathtt{R}\mathtt{R}^{*\top})$ and the translation error as $\Delta\mathbf{t} = \|\mathbf{t} - \mathbf{t}^*\|_2$, where $\mathtt{R}^* \in SO(3)$ and $\mathbf{t}^* \in \mathbb{R}^3$ are the ground truth camera pose of the query image available in the benchmark [3].

Results of 3D point recovery As shown in Fig. 6(a) sphere cloud achieves significantly higher geometric errors compared to ULC [35] and PPL/PPL+ [16] due to its ability to neutralize the geometry-revealing attack [5]. This pattern is repeated in Fig. 6(b) where the sphere cloud shows the lowest image quality due to large geometric errors. Also, Fig. 7(f) shows that no content is revealed using the 3D points estimated from the sphere cloud.

Direct image reconstruction about the map centroid We tried to assess the sphere cloud's resilience to a new type of attack based on direct image-synthesis about the sphere centre. For this purpose, we rotated the camera viewpoint about the sphere centre (map centroid) and projected sphere points to a virtual image plane for image reconstruction via InvSfM. Since no ground truth is available for these synthesized views, we used the pseudo-GT in Fig. 8 for evaluation as described in Table 1. For ULC [35] and PPL/PPL+ [16], we used the recovered 3D points using [5] to reconstruct these images. As shown in the same table, the sphere cloud achieves relatively high privacy-preserving ability against this new attack on 12-Scenes which is qualitatively verified in Fig. 8. However, sphere cloud surprisingly underperforms in the 7-Scenes [33] dataset. While this requires further investigation, we anticipate this is partly due to noisy pseudo-GT images in 7-Scenes as observed in Fig. 8.

Localization results Table 2 presents the overall performance of different localization methods including DVLAD+R2D2(+D) [13, 29, 37] and DSAC*(+D) [2] both of which serve as baselines for evaluating depth-guided approaches. Notably, sphere cloud with η=33% runs real-time unlike ULC and PPL/PPL+ as the result of being able to use the p3P solver [25]. On the downside, we observe reduced camera localization accuracy compared to image-based methods, and a slight increase in translation errors when compared to other depth-guided methods. Among the depth-guided methods, sphere cloud exhibits the lowest median rotational errors on both datasets while DSAC* (trained with rendered depth maps) achieves the lowest translation errors. Overall, the sphere cloud shows efficient runtime with a slight reduction in translation accuracy compared to other depth-guided methods.

Table 2: Comprehensive comparison of localization performance across different visual localization methods which are categorized into two groups: *Image-based* and *depth-guided*. The median error of rotation (ΔR) [°], translation (Δ**t**) [cm] and the ratio (%) of the query images localized within each rotation and translation threshold are reported as in [3]. Runtime [ms] includes the whole iterations of LO-RANSAC [7, 15] and non-linear refinement. $\star$ indicates that results are from the official code in [3] and runtime of [13] is not reported in [3]. Note, oracle denotes results of depth oracle (z-oracle) in sphere cloud . **Bold** indicates the best result in each category.

		Image-based localization				Depth-guided localization					
Dataset	Metric	Point cloud [3]	ULC [35]	PPL [16]	PPL+ [16]	DVLAD* +R2D2(+D)[3]	DSAC* (+D)[3]	Sphere (η=25%)	Sphere (η=33%)	Sphere (oracle) (η=25%)	Sphere (oracle) (η=33%)
12-Scenes [33]	ΔR(°) (↓)	0.139	**0.159**	0.170	0.168	0.389	0.397	0.300	**0.288**	0.240	**0.232**
	Δt (cm) (↓)	0.627	**0.727**	0.775	0.765	0.931	**0.735**	1.310	1.282	0.601	**0.577**
	ΔR<3° (%) (↑)	100.0	**100.0**	**100.0**	**100.0**	99.73	**99.98**	99.00	99.34	99.90	**100.0**
	Δt<3cm (%) (↑)	97.94	**95.88**	95.16	95.13	97.06	**99.21**	86.97	87.86	97.22	**97.60**
	Runtime(ms) (↓)	3	96	**91**	**91**	-	84	48	**24**	22	**13**
7-Scenes [33]	ΔR(°) (↓)	0.174	**0.201**	0.206	0.207	0.966	0.655	0.438	**0.405**	0.262	**0.255**
	Δt (cm) (↓)	0.493	**0.613**	0.647	0.647	2.857	**1.573**	2.119	2.051	0.459	**0.443**
	ΔR<3° (%) (↑)	100.0	**100.0**	**100.0**	**100.0**	96.11	**99.05**	97.00	97.58	99.86	**99.93**
	Δt<3cm (%) (↑)	99.85	**99.32**	99.12	98.96	55.90	**82.81**	69.75	70.93	98.21	**98.51**
	Runtime (ms) (↓)	3	82	**78**	79	-	80	52	**25**	31	**16**

We also compared the localization performance of the *depth-oracle* case of the sphere cloud. Due to the reduced noise in depth measurements, we observed improvements in the localization accuracy of the sphere cloud in oracle cases, likely from the accuracy of the solutions obtained by the p3P solver [25]. Surprisingly, the oracle cases of the sphere cloud show competitive localization accuracy compared to ULC [35] and PPL/PPL+ [16] and significantly outperform them in runtime due to the allowance of the efficient p3P solver. Hence, we anticipate further research of reducing noises on depth measurements will improve the localization accuracy of the sphere cloud.

5 Conclusion and limitations

In this work, we presented a new privacy-preserving scene representation called *sphere cloud* which can nullify the known geometry-revealing attack for line clouds. We noted the main challenges in realizing this representation, namely the possibility of a new type of attack directly revealing images from the sphere points about the map centroid and the issue of unresolved translation scale. We addressed these issues by introducing fake points with recycled real descriptors to thwart direct image reconstruction and presenting an efficient RGB-D privacy-preserving localization framework to guide the translation scale. Experimental results showed that sphere cloud successfully neutralizes the known geometry attack and gains resilience to a new direct attack while reporting around 20–30 fps localization speed. This demonstrates its potential as an efficient privacy-preserving scene representation.

Out of many limitations, our framework exhibits lower translation accuracy due to noisy depth measurements. We also observe the trade-off between localization accuracy and privacy-preserving performance when the proportion of true positive sphere points (η) changes and we have not outlined a principled approach to setting η along with other hyperparameters (*e.g.* σ) to yield optimal performance. We leave improvements to these for future work.

Acknowledgement This work was supported by the NRF (National Research Foundation of Korea) grants funded by the Korea government (MSIT) (No. 2022R1C1C1004907).

References

[1] Clemens Arth, Daniel Wagner, Manfred Klopschitz, Arnold Irschara, and Dieter Schmalstieg. Wide area localization on mobile phones. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 73–82. IEEE, 2009.

[2] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *TPAMI*, 2021.

[3] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6218–6228, October 2021.

[4] Robert Castle, Georg Klein, and David W Murray. Video-rate localization in multiple maps for wearable augmented reality. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 15–22. IEEE, 2008.

[5] 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 (CVPR)*, pages 15663–15673, 2021. doi: 10.1109/CVPR46437.2021.01541.

[6] 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*, pages 13132–13141, 2023.

[7] Ondrej Chum, Jirí 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.

[8] Deeksha Dangwal, Vincent T Lee, Hyo Jin Kim, Tianwei Shen, Meghan Cowan, Rajvi Shah, Caroline Trippel, Brandon Reagen, Timothy Sherwood, Vasileios Balntas, et al. Mitigating reverse engineering attacks on local feature descriptors. In *Proceeding of the British Machine Vision Conference (BMVC)*, 2021.

[9] Yaqing Ding, Jian Yang, Viktor Larsson, Carl Olsson, and Kalle Åström. Revisiting the p3p problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4872–4880, 2023.

[10] Mihai Dusmanu, Johannes L. Schönberger, Sudipta N. Sinha, and Marc Pollefeys. Privacy-preserving image features via adversarial affine subspace embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14262–14272, 2021. doi: 10.1109/CVPR46437.2021.01404.

[11] Marcel Geppert, Viktor Larsson, Johannes L Schönberger, and Marc Pollefeys. Privacy preserving partial localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17337–17347, 2022.

[12] Richard Hartley. Cheirality. *International journal of computer vision*, 26(1):41–61, 1998.

[13] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv preprint arXiv:2007.13867*, 2020.

[14] Tong Ke and Stergios I Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7225–7233, 2017.

[15] Viktor Larsson. PoseLib - Minimal Solvers for Camera Pose Estimation. https://github.com/vlarsson/PoseLib, 2020. Accessed: 2022-10-30.

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

[17] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[18] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, volume 1, page 1, 2015.

[19] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[20] 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 (CVPR)*, pages 9773–9783, June 2024.

[21] Heejoon Moon, Jongwoo Lee, Jeonggon Kim, and Je Hyeong Hong. Supplementary document of depth-guided privacy-preserving visual localization using 3d sphere clouds, 2024.

[22] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31 (5):1147–1163, 2015.

[23] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[24] Linfei Pan, Johannes Lutz Schönberger, Viktor Larsson, and Marc Pollefeys. Privacy Preserving Localization via Coordinate Permutations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023.

[25] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (p3P) solver. In *Proceedings of the European conference on computer vision (ECCV)*, pages 318–332, 2018.

[26] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. Segloc: Learning segmentation-based representations for privacy-preserving visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15380–15391, 2023.

[27] Francesco Pittaluga, Sanjeev Koppal, and Ayan Chakrabarti. Learning privacy preserving encodings through adversarial training. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 791–799. IEEE, 2019.

[28] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 145–154, 2019.

[29] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32, 2019.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.

[31] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy preserving visual SLAM. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 102–118. Springer, 2020.

[33] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013.

[34] Zhenbo Song, Wayne Chen, Dylan Campbell, and Hongdong Li. Deep novel view synthesis from colored 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 1–17. Springer, 2020.

[35] 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 (CVPR)*, pages 5493–5503, 2019.

[36] Henrik Stewénius, Magnus Oskarsson, Kalle Aström, and David Nistér. Solutions to minimal generalized relative pose problems, 2005.

[37] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1808–1817, 2015.

[38] Julien Valentin, Angela Dai, Matthias Niessner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 323–332, 2016. doi: 10.1109/3DV.2016.41.

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.