# PIPE: Privacy-preserving 6DoF Pose Estimation for Immersive Applications

Nan Wu
George Mason University
Fairfax, VA, USA
nwu5@gmu.edu

Ruizhi Cheng
George Mason University
Fairfax, VA, USA
rcheng4@gmu.edu

Songqing Chen
George Mason University
Fairfax, VA, USA
sqchen@gmu.edu

Bo Han
George Mason University
Fairfax, VA, USA
bohan@gmu.edu

## Abstract

Image-based mapping and localization offer six degrees of freedom (6DoF) pose estimation for immersive applications. This is achieved by matching, on a server, 2D visual features extracted from a mobile device's camera view and 3D features stored in a map. While effective, this process may lead to privacy breaches (*e.g.,* exposure of sensitive information captured by camera views). To tackle this crucial issue, we present PIPE, a first-of-its-kind *P*rivacy-preserving *I*mage-based 6DoF *P*ose *E*stimation system. The design of PIPE is motivated by our key observation that *uploading only a small amount of features extracted from camera views for pose estimation could reduce privacy leakage.* However, trade-offs exist between privacy preservation, system utility (*i.e.,* pose estimation accuracy), and system performance (*e.g.,* end-to-end latency). To balance the trade-offs, PIPE deliberately explores the feature-detection space to reduce computation latency, designs an efficient feature ranking method by judiciously utilizing map data, and optimizes feature selection by jointly considering the features' ranking and spatial distribution to improve pose estimation accuracy. Moreover, we construct a learning-based metric to quantify the extent of privacy leakage in images. Our extensive performance evaluation reveals that PIPE can effectively preserve privacy and reduce end-to-end latency by up to 22.6%, while marginally affecting pose estimation accuracy (*e.g.,* as low as 2.7%).

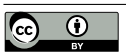
## CCS Concepts

• **Computing methodologies** → **Mixed / augmented reality**; **Computer vision**; **Computer vision**; **Mixed / augmented reality**; • **Human-centered computing** → **Mobile computing**; **Mobile computing**; • **Security and privacy** → **Privacy protections**; **Privacy protections**.

## Keywords

Mobile Spatial Computing, 3D Spatial Maps, Image-based Localization, and Privacy Leakage

**ACM Reference Format:**
Nan Wu, Ruizhi Cheng, Songqing Chen, and Bo Han. 2025. PIPE: Privacy-preserving 6DoF Pose Estimation for Immersive Applications. In *The 23rd ACM Conference on Embedded Networked Sensor Systems (SenSys '25), May 6–9, 2025, Irvine, CA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3715014.3722069




*SenSys '25, May 6–9, 2025, Irvine, CA, USA*

ACM ISBN 979-8-4007-1479-5/2025/05
https://doi.org/10.1145/3715014.3722069

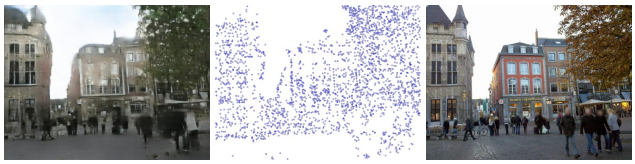


**Figure 1: Reconstructed image (left) from extracted features (middle) closely resembles the ground truth (right).**

## 1 Introduction

Emerging immersive technologies, such as augmented and mixed reality (AR/MR), heavily rely on innovations in image-based spatial mapping [16, 89] and localization [86] for six degrees of freedom (6DoF) pose estimation of mobile devices [45, 54, 67, 70]. Take Google's ARCore Geospatial [5] as an example. It extracts visual features from street view images and generates a 3D spatial map of the environment on a server. During localization, a mobile device uploads its camera view (*i.e.,* localization image) to the server, which extracts the same type of 2D features as those in the 3D map and performs feature matching to estimate the device's 6DoF pose (*i.e.,* position and orientation).

Image-based pose estimation raises privacy concerns for users, as localization images may contain sensitive information, particularly when used in home environments or confidential industrial settings [47, 96]. A straightforward solution is to have mobile devices extract visual features from localization images and send them, instead of raw images, to the server for pose estimation. However, existing work [35, 39, 66, 104] has revealed that attackers could reconstruct original images with high fidelity solely from extracted features, as shown in Figure 1. This poses a risk of privacy leakage, as the disclosed information in scenes (*e.g.,* pedestrian activities) can affect subjective perceptions of privacy (§2). Although the computer vision (CV) community recently proposed several methods [40, 73, 96] for protecting extracted features by concealing their positions or descriptors, it is still possible to restore the original images (§2).

In this paper, we propose PIPE, an innovative end-to-end system, to preserve privacy in localization images for 6DoF pose estimation of mobile devices. Given the *subjective* nature of privacy [75, 94], especially in the context of images, the overarching goal of PIPE is to diminish the leakage of information, thereby reducing the risk of privacy breaches. To achieve this formidable goal, PIPE carefully selects a limited subset of extracted features from localization images and sends only them to the server for pose estimation. We discuss the potential implications of this strategy in §7. Existing privacy-preserving techniques, such as fully homomorphic encryption (FHE) [46] and multi-party computation (MPC) [49], are

impractical for image-based pose estimation due to FHE's high computation latency [105] and MPC's communication overhead [32].

Our main insight is that the visual quality of reconstructed images, and thus the leakage of information/privacy, could be drastically reduced by uploading only a small subset of selected features for pose estimation (§2.2). However, blindly doing this will inevitably lower pose estimation accuracy and increase end-to-end latency. Thus, we should *effectively balance the trade-offs between privacy preservation, pose estimation accuracy, and end-to-end latency*. Specifically, the design of PIPE poses the following key challenges. (1) While extracting more features on mobile devices can enhance pose estimation accuracy, it leads to higher overall latency. (2) Without accessing map data stored on the server, assessing the importance of features for pose estimation on mobile devices is challenging. (3) High-ranking features often cluster in visually distinctive areas such as corners, and selecting only a small subset of them can decrease pose estimation accuracy due to reduced spatial diversity. (4) There is no well-celebrated standard metric for evaluating the effectiveness of privacy-preserving image-based 6DoF pose estimation. To address the above challenges, PIPE incorporates the following creative solutions into a holistic system.

**Time-efficient Feature Extraction (§4.1).** Extracting features from the default detection space leads to high computation overhead on mobile devices. Thus, we explore the trade-off between feature extraction time and pose estimation accuracy to properly limit the feature-detection space, dramatically reducing feature extraction time with a marginal impact on pose estimation accuracy.

**Lightweight Model for Feature Ranking (§4.2).** Different features have varied contributions to pose estimation accuracy, depending on their matchability to those in the map. Thus, instead of ranking features purely based on factors such as their scores calculated during extraction [65, 84], PIPE judiciously utilizes map data to train a lightweight feature-ranking model for selecting the important features that contribute to accurate pose estimation.

**Effective Optimization of Feature Selection (§4.3).** We resort to the following insight from pose estimation algorithms to optimize feature selection. High-ranking features, which are likely to match those in the map, may be clustered together, limiting their collective contribution to accurate pose estimation. Thus, a key optimization is to leverage feature positions and deliberately select dispersed high-ranking features to improve pose estimation accuracy.

**New Metric for Measuring Privacy Leakage (§4.4).** Traditional metrics such as the structural similarity index measure (SSIM) [102] predominantly gauge the leakage of information rather than privacy in images. To overcome this limitation, we conduct a comprehensive IRB-approved user study with 360+ participants to understand privacy leakage in reconstructed images and introduce a novel metric that incorporates four basic metrics to quantify the extent of privacy leakage in images. This new metric has a higher correlation with the privacy leakage rated by users than the basic ones.

We build a prototype of PIPE (§5) and thoroughly evaluate its performance via publicly available datasets (§6). We highlight our evaluation results as follows.

• PIPE effectively preserves privacy with limited leakage. With a privacy-leakage level defined from 1 (no privacy leakage) to 10 (high privacy leakage), PIPE leads to a leakage level of merely 1.66 and 1.70 for outdoor and indoor scenarios, respectively. For comparison, without protection, the privacy-leakage level is 7.15 and 7.88 for outdoor and indoor scenarios, respectively.

• PIPE maintains comparable pose estimation accuracy as the baseline (*i.e.,* with no privacy protection). On a large-scale dataset [88], the 75th percentile of pose estimation errors for PIPE (baseline) is about 0.20m (0.17m) for position and 0.39° (0.31°) for orientation.

• By selecting and uploading fewer features to the server, PIPE significantly reduces computation overhead, decreasing the end-to-end latency by up to 22.6% compared with the baseline (*i.e.,* sends images to the server for pose estimation).

The novelty of PIPE lies in reducing information and privacy leakage by selecting only the indispensable features, thereby maintaining high precision in 6DoF pose estimation and safeguarding user privacy. While feature selection has been utilized to preserve privacy in machine learning, such as leveraging gradient-based perturbation to identify essential features for accurate model prediction [69], it is unsuitable for PIPE. This is because, after feature extraction, the core steps of state-of-the-art 6DoF pose estimation schemes [55, 85] do not utilize machine-learning models, as existing learning-based methods are still less accurate [32, 38], making the above approach ill-suited. However, with learning-based localization methods continuing to advance, future extensions of PIPE could incorporate neural networks to identify the most relevant learned features for pose estimation while avoiding the transmission of less informative ones.

## 2 Background and Motivation

### 2.1 Background

**Spatial Map Construction.** Structure from motion (SfM) [16, 89] is a widely adopted technique for constructing 3D spatial maps with a set of 2D images. It first extracts 2D visual features from images, which usually contain their *position* in the image, a *score* describing how strong they are (*e.g.,* representing contrast response in scale-invariant feature transform (SIFT) features [65]), and a unique *descriptor* that is highly invariant to the image's scale, translation, and rotation, and robust to changes of illumination and viewpoint [37, 65, 83]. SfM then matches these features to identify overlapping images, estimates image poses, triangulates features' *3D positions*, and uses bundle adjustment [99] to reduce errors. We refer to the images for constructing maps as *map images*.

**Image-based Localization for 6DoF Pose Estimation.** Image-based localization determines the position and orientation of mobile devices (*i.e.,* their 6DoF pose) by comparing images that they capture to a spatial map [87]. We refer to these images as *localization images*. Image-based localization first extracts features from localization images and matches them with those in the map based on the Euclidean/Hamming distance between feature descriptors. Then, it calculates the device's 6DoF pose by performing perspective-$n$-point (P$n$P) RANSAC [44] with the positions of the set of $n$ matched features. We refer to features from localization images that contribute to pose estimation as *inlier features*. SIFT is commonly used for image-based mapping and localization [16, 86, 89].

**Privacy Leakage of Image-based Pose Estimation.** With the development of cloud-based services, large-scale 6DoF localization can

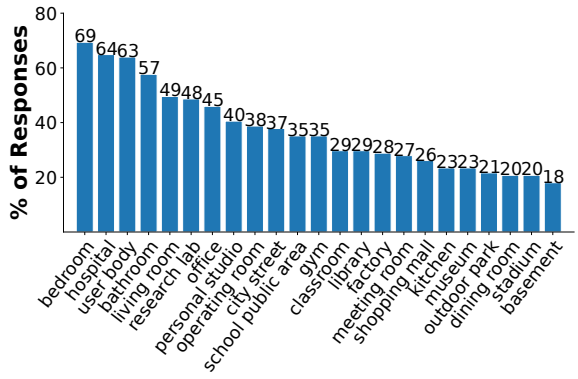



**Figure 2: Common scenarios of AR/MR applications with the percentage of users having privacy concerns.**

be achieved for mobile devices [5, 20, 76]. Compared with schemes such as GPS, image-based solutions are more accurate [18, 86] and provide both position and orientation for mobile devices, offering more opportunities for AR/MR [3, 8, 29, 60]. However, it requires sending images to the server, which may contain sensitive visual data, leading to privacy leakage when using untrusted localization services or under eavesdropping adversary attacks [1, 2, 74, 96].

Instead of directly sending localization images, an alternative is to retrieve features from them and send extracted features to the server. However, this approach is not exempt from potential privacy threats, as attacks employing convolutional neural networks (CNN) can reconstruct original images from the features [35, 39, 66, 78, 104]. For example, Dosovitskiy *et al.* [39] proposed an encoder-decoder CNN for reconstructing images from extracted features. More recently, the introduction of generative adversarial networks (GAN) and advanced CNN models has further enabled the recovery of high-quality images from extracted features [35, 78, 104].

## 2.2 Motivation

**Privacy Concerns in Localization Images for Pose Estimation.** To understand privacy concerns related to mobile devices capturing images for AR/MR experiences, we conducted an IRB-approved study through the Prolific platform [81]. We asked 111 participants (52.3% male, 45.0% female, 2.7% unspecified) about their privacy concerns across 23 common AR/MR scenarios [7] and let them identify objects that trigger concerns. In Figure 2, we present the scenarios in descending order of the percentage of user responses expressing privacy concerns. The results highlight the need to prevent privacy leakage across all scenarios. Moreover, the varied privacy preferences emphasize the importance of *user-centric privacy control* that can adjust the level of privacy preservation in response to different preferences (*i.e.,* lower privacy protection for higher system utility).

**Limitations of Existing Techniques.** There are two main approaches to reduce privacy leakage from localization images. The first protects feature positions. For example, Speciale *et al.* [96] convert extracted features to random lines before sending them to the server, which hides the content of original images but preserves the positional relationships of features necessary for 6DoF pose estimation. However, this method remains susceptible to reconstruction attacks similar to those used by Chelani *et al.* [28] on 3D data, which recover original point positions from the closest points between pairs of lines. The second approach protects feature descriptors. For instance, NinjaDesc [73] utilizes an adversarial training network to transform original descriptors into altered values, degrading image reconstruction quality while maintaining matching capabilities. However, a recent study by Wu *et al.* [104] demonstrates that privacy can be compromised by reconstructing images using only feature positions.

**Figure 3: Reconstructed images from 100% (left), 50% (middle), and 10% (right) of extracted features. The image is highly blurred with only 10% of the features.**

| Method | # of Matched Features | # of Inlier Features | Recall @ Thresh. (%) | | |
|---|---|---|---|---|---|
| | | | High | Med. | Low |
| All | 501±408 | 353±372 | 83.4 | 91.3 | 96.0 |
| 10% | 69.7±46.4 | 38.6±38.5 | 68.5 | 77.8 | 85.5 |

**Table 1: Pose estimation accuracy and the number of matched and inlier features using all extracted and 10% of randomly selected SIFT features on the Aachen dataset [88]. On average, there are 2,727 features extracted from images. We show the recalls of accurate pose estimation under high, medium, and low-precision intervals.**

**Trade-offs between Privacy and Utility.** Prior work [35, 105] shows that randomly reducing features from localization images degrades the quality of reconstructed images. Our experiments on the Aachen dataset [88] show highly blurred images with 10% of randomly selected SIFT features, and details are significantly diminished (Figure 3). However, random feature selection reduces matching utility [35] and increases pose estimation error [105].

To analyze the pose estimation accuracy, we compare position and orientation errors using all features versus 10% of randomly selected features on the Aachen dataset [88]. The position error is calculated as the Euclidean distance between the ground truth and the estimated position, while the orientation error measures the minimum rotation-angle difference [51]. Following the common practice [88], we report the recall of pose estimation with position errors within $X$ meters and orientation errors within $Y$ degrees under high-precision ($X$ as 0.25, $Y$ as 2), medium-precision ($X$ as 0.5, $Y$ as 5), and low-precision ($X$ as 5, $Y$ as 10) intervals. As shown in Table 1, using 10% of randomly selected features leads to a 14.9% accuracy drop (from 83.4% to 68.5%) for high-precision pose estimation. This is because only around 13% of all the extracted features are inliers, and randomly dropping 90% of features results in losing approximately 90% of inliers for accurate pose estimation.

The above issues inspire us to design a lightweight method that carefully selects a subset of features with a high likelihood of passing the matching step and being identified as inliers by the P$n$P RANSAC algorithm in pose estimation. This approach preserves privacy by limiting exposed features while maintaining pose estimation accuracy by including more inliers than random selection.

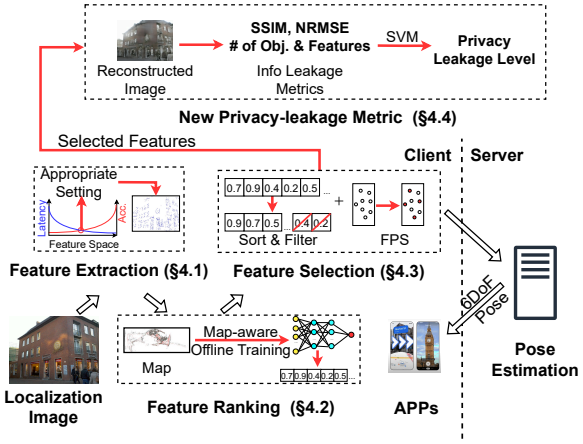



**Figure 4: System architecture and workflow of PIPE, a generic solution that enhances pose estimation with privacy protection by selecting a subset of features without modifying the existing pipeline.**

# 3 Overview and Threat Models

## 3.1 Overview

PIPE aims to strike a balance between privacy protection, system utility (*i.e.,* pose estimation accuracy), and system performance (*i.e.,* end-to-end latency for pose estimation). As shown in Figure 4, to achieve this goal, PIPE extracts features from localization images with a constricted feature-detection space (§4.1) and uploads only a critical subset of extracted features that are essential for pose estimation. The key challenge is to ensure the selected features still carry enough information for accurate pose estimation. PIPE strategically selects a subset of features, focusing on high-ranking (§4.2) and non-clustered (§4.3) ones to preserve essential data for pose estimation while reducing privacy leakage. Since privacy is often subjective and context-dependent, existing metrics such as SSIM [102] fail to reflect privacy risks from images accurately. To address this, we propose a new learning-based metric that incorporates traditional metrics to capture leakage from various perspectives (§4.4).

Note that it is prohibitive to continuously perform pose estimation on mobile devices without a server, due to its high computation overhead and the large size of spatial maps. For example, executing simultaneous localization and mapping (SLAM), which has been utilized in ARKit [6] and ARCore [4], on mobile devices continuously can be challenging because of the increasing complexity as the map grows [24]. Aligning with the practical localization in large-scale environments [68], PIPE is not designed to localize every image in real-time. Instead, lightweight SLAM can be performed locally for small-scale tracking, reducing computation and memory usage, and PIPE focuses on the localization of keyframe images selected by SLAM. These images are periodically sent to a server to utilize its enhanced storage and processing capabilities for localization. This ensures accurate localization when transitioning into a new area within a global map or correcting accumulated errors in local tracking. Another practical system design involves storing the entire map on a server and retrieving relevant subsets based on GPS locations to reduce on-device computation and memory usage for pose estimation. However, map owners may not want to share their data with other parties [32].

## 3.2 Threat Models

**Attack Scenarios.** We examine two predominant adversarial scenarios. The first considers the semi-honest security model [50, 100], where involved participants, such as the server, are not trusted with secrets but are expected to adhere to the prescribed pose estimation protocol [17, 27, 106]. Attackers with access to the server, such as privileged employees, can obtain user-uploaded features and all internal values in pose estimation. Their privileged status allows them to bypass traditional security checks, making their activities hard to be detected, and thus more concerning. The second scenario is the eavesdropping adversary [43, 96, 97], specifically in the wireless context, where an attacker deftly taps into the communication (uploaded features) between the client and the server. This attacker's interception, often through a man-in-the-middle attack, gains access to extracted features from localization images. In both scenarios, adversaries aim to infer private information without alerting the system or users, and unauthorized access could unravel users' sensitive surrounding environments and activities.

**Attack Methods.** Attackers infer privacy from extracted features for pose estimation by first applying reverse-engineering techniques on the features to reconstruct images that closely resemble the original ones [35, 73, 96, 104]. Such reconstructions can expose sensitive information, thereby increasing privacy concerns. For a more efficient and automated privacy inference than what manual analysis offers, after attempting to re-generate original images from features, attackers subsequently employ machine learning techniques to further extract or deduce private information from images (*e.g.,* via object detection) [35].

**Privacy Concerns for Maps and Pose Estimation Results.** While our focus centers on localization images, we acknowledge potential privacy concerns regarding spatial maps and pose estimation results. However, these issues are beyond the scope of PIPE. We assume that users upload spatial maps with privacy-sensitive objects removed and use the pose estimation service with full knowledge and consent.

# 4 System Design of PIPE

## 4.1 Time-efficient Feature Extraction

**Problem and Challenges.** The first step in PIPE is to extract features from localization images. While having more features can improve pose estimation accuracy, this process is computationally intensive on mobile devices, and the end-to-end latency grows with the number of extracted features. To address this problem, the feature detection space should be constricted to reduce feature-extraction time. However, this constriction may negatively affect pose estimation accuracy due to the decreased number of extracted features. The key challenge lies in striking the right balance between computational efficiency and pose estimation accuracy.

**Our approach.** To optimize feature-detection space with a limited negative impact on pose estimation accuracy, PIPE explores various configurations of feature-searching layers and image resolution to accelerate feature extraction without sacrificing pose estimation accuracy. *The joint consideration of latency and accuracy sets our*

| Method | Recall @ Thresh (%) | | |
|---|---|---|---|
| | High | Med. | Low |
| Default SIFT | 83.6 | 92.0 | 97.6 |
| Rmv 1st Octave | 81.1 | 89.9 | 95.8 |
| 1 Octave Layer | 83.4 | 91.3 | 96.0 |

**Table 2: Pose estimation results with images at 1024×768 under three SIFT settings.**

| Resolution | # of Features | # of Inliers | Recall @ Thresh (%) | | |
|---|---|---|---|---|---|
| | | | High | Med. | Low |
| 1600×1200 | 13,938±4,875 | 1,333±1,450 | 90.5 | 94.7 | 98.3 |
| 1024×768 | 5,030±1,744 | 739±740 | 90.6 | 94.7 | 98.2 |
| 768×576 | 2,951±1,008 | 477±466 | 87.4 | 93.2 | 97.4 |
| 512×384 | 1,551±500.8 | 254±251 | 85.7 | 91.7 | 95.7 |

**Table 3: Pose estimation accuracy and the number of features and inliers under the resolutions of 1600×1200, 1024×768, 768×576, and 512×384.**

*method apart from traditional CV techniques that primarily focus on accuracy alone.* We use SIFT [65] as a case study to illustrate PIPE's design, since it is robust and effective for state-of-the-art image-based mapping and localization [47, 78, 89, 95].

***Limiting Feature-searching Layers.*** SIFT [65] starts by constructing octaves, with the first octave created by upsampling the original image. Successive octaves are initiated by images downsampled by a factor of two from the previous octave until the image size is too small for further downsampling. In each octave, SIFT generates increasingly blurred images using a Gaussian kernel and computes pixel-wise differences between successive ones to create images of difference of Gaussians (DoG). These DoG images form the layers of an octave where features are detected, with the number of layers being user-defined (*e.g.,* the default is 3 in OpenCV [26]).

We explore two strategies to reduce feature-extraction overhead. First, since most SIFT features are detected in the first octave, excluding them can significantly reduce extraction time. Second, restricting the number of layers in each octave (*e.g.,* from the default 3 to 1) reduces the search space and speeds up detection. While both strategies decrease extraction time, they have different impacts on pose estimation accuracy. Table 2 illustrates our experiments on the Aachen [88] dataset. Removing first-octave features results in a 2.5% accuracy drop at high-precision intervals (from 83.6% to 81.1%). In contrast, limiting octave layers marginally impacts pose estimation accuracy, especially for the high-precision interval (from 83.6% to 83.4%). Thus, PIPE constrains octave layers for feature extraction. This approach can be extended to other algorithms, such as speeded-up robust features (SURF) [23], similar to SIFT for using scale-invariant features.

***Image Resizing.*** Feature extraction takes a longer time to process higher-resolution images with more pixels. However, compared with lower-resolution images, they may not offer significantly more information for accurate pose estimation. We conduct experiments on Samsung Galaxy S22+ with the Aachen [88] dataset to understand the trade-off between pose estimation accuracy and feature-extraction time for different resolutions. We select localization images with 1600×1200 resolution and resize them to 1024×768, 768×576, and 512×384.

Table 3 shows the accuracy of pose estimation with different resolutions under three intervals. The results indicate that resizing images from 1600×1200 to 1024×768 reduces feature extraction time by over 50% (from 463±27 ms to 219±16 ms) with minimal impact on accuracy. However, further reduction in resolution decreases accuracy. Similar results are observed with other outdoor and indoor datasets, including the GreatCourt [58], the 7Scenes [92], and NYU [72] datasets. Thus, PIPE resizes images to 1024×768 to balance efficiency and pose estimation accuracy.

## 4.2 Lightweight Model for Feature Ranking

**Problem and Challenges.** Not all features are equally important for 6DoF pose estimation. As shown in §2.2, only around 13% of extracted features are inliers, thereby making a limited contribution to accurate pose estimation. Hence, PIPE needs to identify the subset of extracted features that hold greater importance for pose estimation. Unlike server-side feature matching and P*n*P RANSAC, which leverage spatial maps to choose features, the challenge of inlier-feature selection for a PIPE client is to filter out outliers without the resource-intensive matching with map features.

**Our approach.** Our key insight is that we can interpret feature selection as classifying whether a feature matches any map features. This is due to the similarities between feature matching in pose estimation and classification. Specifically, in pose estimation, two features with similar descriptors are matchable, while in classification, instances are grouped into the same class if they exhibit sufficient similarity. Based on this insight, PIPE trains a lightweight binary classification model using a multilayer perceptron (MLP) [10] that determines if image features are matchable to map features based on their descriptors. The model outputs confidence scores that indicate their matchability. Considering that the features are extracted from images, applying MLP to these features aligns with common practices in CNN, where the final layers typically leverage MLPs to produce classification results.

We utilize the features from all map images for training, labeling them as inliers or outliers based on whether they contribute to the map. The ranking model is lightweight and plug-and-play, allowing for seamless integration with other feature descriptors (*e.g.,* ORB [84] as shown in §6). Specifically, a fundamental aspect of our design is the effective use of map data for training from a system perspective. This design considers the trade-off between accuracy and privacy, as our goal is to improve localization accuracy and reduce privacy leakage by reducing the number of uploaded features while ensuring they remain essential for localization. As a result, the ranking model is designed to leverage map data to optimize feature selection based on a given map, retaining only the most relevant features while filtering out less informative ones as much as possible.

While our proposed model is map-specific, we can employ transfer learning [77] for rapid model training. The reason is that features from different maps may exhibit shared structural and characteristic similarities. For example, in urban areas, buildings often possess similar structures irrespective of their geographical location. These

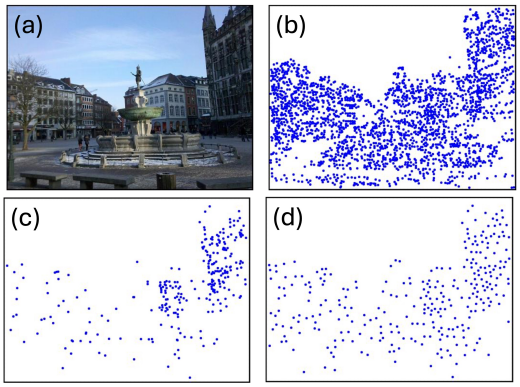



**Figure 5: Comparison of top 10% ranked features (c) and those determined by PIPE's farthest point sampling (d), alongside the original image (a) and all features (b).**

shared structures enable the application of transfer learning, allowing for the knowledge acquired from one map to be effectively utilized in others, considerably accelerating training (§6.4).

Note that our proposed solution is different from methods for reducing descriptor dimensions, such as t-distributed stochastic neighbor embedding (t-SNE) [12] and Lasso [9]. The reason is that they do not protect feature positions and can inadvertently reduce the distinctiveness of features, which is crucial for accurate feature matching and pose estimation.

## 4.3 Effective Optimization of Feature Selection

**Problem and Challenges.** Suppose that PIPE needs to select and send $p$% features for pose estimation. While ranking features based on matchability can improve accuracy, selecting only high-ranking features may reduce spatial diversity since they often cluster in visually distinctive areas such as corners, where the identifiable characteristics of these regions enhance matching accuracy. This reduced spatial diversity may make accurate pose estimation challenging. For example, when using four pairs of matched points for P3P RANSAC, clustered points might be treated as a single point in the worst case, offering minimal positional constraints. Thus, the challenge lies in selecting high-ranking features that jointly contribute the most to accurate pose estimation.

**Our approach.** Our key observation is that, for robust and accurate pose estimation, the selected features should not only be inliers but also be as scattered as possible. This is because well-distributed features cover a larger area and different parts of the scene, better conditioning the calculations involved in the P$n$P RANSAC algorithm. Therefore, we design a filter inspired by the farthest point sampling (FPS) algorithm [42]. For ranked features, we first filter out those with a confidence score lower than a threshold (*e.g.,* 0.5, a standard threshold used in the sigmoid function for binary classification [53]). Then, we select the $p$% of features via FPS from the remaining ones by leveraging only their 2D positions. For certain images, there may be less than $p$% of features with a confidence level higher than 0.5. Thus, PIPE begins by examining the confidence scores of the top $p$% of sorted features. If any score is below the defined threshold, we directly select the top $p$% of ranked features.

Figure 5 shows an example with a localization image. Selecting the top 10% of ranked features results in pose estimation errors of 0.5168 m for position and 0.4949° for rotation. However, our method makes the selected features scattered, providing more features on the left side of the image for more robust pose estimation. Thus, the resulting position and rotation errors are 0.0334 m and 0.0987°, close to the results with all features (0.0328 m and 0.0637°).

## 4.4 New Privacy Leakage Metric

**Problem and Challenges.** There is no well-celebrated metric for quantifying the effectiveness of privacy-preserving schemes for images. Prior works [35, 73] utilize SSIM [102] to demonstrate the efficiency of privacy preservation by comparing the similarity between the original and reconstructed images. However, privacy is often subjective [75, 94] and context-dependent. A low similarity does not necessarily equate to less privacy leakage. Although differential privacy [41] is widely used and provides a level of assurance, it may not be suitable for image-based pose estimation (§7).

**Our approach.** We propose a data-driven metric, named pLEAK, that utilizes measures from various perspectives as input to comprehensively quantify privacy leakage. pLEAK evaluates privacy through a human-centric perspective, reflecting subjective interpretations of images' privacy leakage. To ensure the generalizability of pLEAK, we collect data from an extensive user study encompassing both indoor and outdoor scenarios that represent real-world cases.

We first enumerate several metrics and analyze how they reflect potential privacy leakage. On the pixel level, SSIM [102] measures the similarity between the original and reconstructed images, where a higher SSIM implies that more original details are preserved, potentially increasing privacy leakage. NRMSE [11] quantifies pixel-wise reconstruction error, and a lower NRMSE corresponds to higher similarity and potentially more privacy leakage. From the content perspective, object detection measures the number of identifiable objects, with fewer objects suggesting better privacy protection. For information leakage, fewer extracted features from the reconstructed images indicate less retrievable information, reducing privacy risks. We also consider information entropy as a candidate metric since it quantifies the distribution of pixel intensities within an image, and reconstructed images that retain details tend to have a higher entropy than those with lost details.

Given that the privacy risk of an image is subjective and is defined by human perception, we conduct an IRB-approved online user study to label privacy-leakage levels for different images, providing the ground truth for training and validating our model. We randomly select 500 images from the outdoor Aachen dataset [88] and another 500 images from the indoor NYU depth dataset [72]. For each localization image, we present users with five other images reconstructed from 10%, 30%, 50%, 70%, and 90% of its features extracted and selected by PIPE. Thus, this user study involves 6,000 images and participants rate privacy leakage on a scale from 1 to 10. Before the online user study, we use anonymized example images with varying levels of sensitive information (*e.g.,* faces and identifiable objects) to help participants understand what may constitute privacy leakage, ensuring common understanding among all participants. To reduce the impact of outliers, we calculate the average rating for each image after removing the highest and lowest ratings.

| Metric | Outdoor | Indoor |
|---|---|---|
| SSIM | 0.604 | 0.583 |
| NRMSE | -0.632 | -0.674 |
| # of Objects | 0.683 | 0.754 |
| # of Features | 0.797 | 0.624 |
| Information Entropy | 0.071 | 0.086 |
| pLEAK (Ours) | **0.850** | **0.909** |

**Table 4: The Spearman correlation coefficients between various metrics and privacy-leakage ratings from user study. Variables with absolute values ≥0.3 are considered correlated.**

Furthermore, we collect at least 10 ratings for images that show high variance in privacy perceptions until the variance becomes stable, ensuring that our results remain robust and reliable.

We then evaluate the correlation between these candidate metrics and the ground-truth privacy leakage ratings obtained from our user study. Table 4 shows the Spearman correlation between the selected metrics and user ratings. SSIM, NRMSE, object count, and feature count all show high correlations (correlation coefficient ≥ 0.3) with the perceived privacy leakage. In contrast, information entropy shows a low correlation with the user ratings and was therefore excluded from the formulation of pLEAK. Specifically, SSIM and NRMSE show similar correlations for both indoor and outdoor images. The number of objects has a much higher correlation for indoor scenarios (0.752) than outdoors (0.686), while the number of features has a much higher correlation for outdoor scenarios (0.801) than indoors (0.620). This is probably because outdoor images offer a greater variety of feature-level information due to diverse content and lighting conditions, whereas indoor images usually contain more privacy-related objects, both of which benefit the measurement of privacy leakage.

We further conduct a cross-correlation analysis among the four candidate metrics, including SSIM, NRMSE, number of objects, and number of features, to evaluate their mutual redundancy. Our results show that all pairwise correlations for outdoor and indoor datasets remain <0.7, showing moderate correlations. This indicates that no two metrics exhibit high redundancy, thereby justifying the inclusion of all four metrics in the formulation of pLEAK.

Since the above metrics (*i.e.,* SSIM, NRMSE, number of objects, and number of features) reflect privacy leakage from different perspectives, pLEAK leverages a learning-based approach that integrates them to comprehensively measure the overall privacy leakage from images. Rather than providing a formal privacy guarantee, pLEAK serves as a proxy for privacy leakage by combining multiple objective metrics to approximate subjective privacy risks based on observable and quantifiable attributes of the reconstructed image. While it does not enforce theoretical privacy protections, pLEAK offers a practical assessment of privacy leakage by estimating the extent of retained information in a way that aligns with human perception. Our method involves training a regression model with the above four metrics as input and producing an output representing the privacy-leakage level, ranging from 1 (no privacy leakage) to 10 (the highest level of privacy leakage). We fine-tune the pre-trained Yolov4 [25] model by including private objects identified in §2.2 and use SIFT [65] for feature extraction.

Acknowledging that individual users may overlook sensitive information, we collect ratings on privacy leakage from a wide range of participants to ensure our metric reflects diverse views. To this end, we have collected responses from an extensive user study with a total of 230 participants from 33 countries (48.3% male, 50.0% female, and 1.7% unspecified). The age distribution spans from 18 to 40+, with 15.2% of participants aged 18–24, 25.2% aged 24–30, 30.0% aged 30–40, and 29.6% over 40. For each image, we collect at least ten responses, resulting in a total of 60,000+ ratings.

Table 4 shows the Spearman correlation between the four metrics and user ratings, confirming their relevance (correlation coefficient ≥ 0.3). SSIM and NRMSE show similar correlations for both indoor and outdoor images. The number of objects has a much higher correlation for indoor scenarios (0.752) than outdoors (0.686), while the number of features has a much higher correlation for outdoor scenarios (0.801) than indoors (0.620). This is probably because outdoor images offer a greater variety of feature-level information due to diverse content and lighting conditions, whereas indoor images usually contain more privacy-related objects, both of which benefit the measurement of privacy leakage.

We train pLEAK with linear regression (LR), support vector regression (SVR), and MLP, and perform five-fold cross-validation. The average prediction errors in outdoor/indoor scenarios are 10.2%/8.6%, 8.3%/7.3%, and 10.8%/10.9%, with LR, SVR, and MLP, respectively. Thus, we select SVR for pLEAK to measure privacy leakage. Table 4 shows the Spearman correlation between the ground-truth privacy-leakage levels and the predicted values are high, 0.850 for outdoor and 0.909 for indoor scenarios. This validates pLEAK's reliability in accurately inferring privacy leakage from images.

## 5 System Implementation

**User-centric Privacy Control.** A key feature of PIPE is that it allows users to choose their desired privacy-protection level, offering high, medium, and low options. To this end, we conduct another IRB-approved online user study to determine the value of $p$ for ensuring different levels of privacy protection. In the study, we randomly select 500 images from the Aachen dataset [88]. We present participants with the original images and those reconstructed from $p$% of features selected by PIPE in descending order, from 100 to 10 at a step of 10. For each image, we ask participants the value of $p$ that is sufficient to preserve privacy.

We collect 10+ responses for each image from a total of 139 participants from 25 countries (57.3% male, 42.0% female, and 0.7% unspecified). For the age distribution, 15.4% of participants are 18 to 24 years old, 25.2% are 24 to 30, 30.1% are 30 to 40, and 29.3% are older than 40. From the 5,000+ ratings collected, we compute the 25th, 50th, 75th, and 95th percentiles, which correspond to different levels of privacy protection. The 25th percentile, representing the threshold at which at least 50% of participants perceived sufficient privacy protection, corresponds to selecting 70% of features. Similarly, the 50th, 75th and 95th percentiles, satisfying 50%, 75% and nearly all users, correspond to selecting 40%, 20% and 10% of features, respectively.

**Implementation.** We develop a prototype of the PIPE server with Linux and the PIPE client on Samsung Galaxy S22+ (Android 13). The PIPE server, which provides pose estimation, incorporates functions from the hierarchical localization (HLoc) toolbox [85]. We

implement all device-side functions with Java8, ensuring compatibility with other Android devices.

Upon receiving localization images, PIPE processes them in Bitmap and utilizes the OpenCV library [26] to resize and convert them to grayscale for feature extraction. Then, PIPE ranks and selects $p$% of extracted features according to users' privacy preferences. The feature ranking model is implemented in Python with TensorFlow [13] and executed on mobile GPU with TensorFlow-lite [64]. The MLP model contains two hidden layers (each with 4,096 neurons), followed by a dropout layer with a rate of 0.1. We train the model with an Adam optimizer [59], setting the batch size as 512, the learning rate as 1e-4, and the training epoch as 50, which is sufficient for convergence.

Our implementation includes over 1,000+ lines of code: 550+ in Java for client operations, and 450+ in Python for server-side pose estimation and model training.

## 6 Performance Evaluation

### 6.1 Experiment Setup

**Mobile Device:** The mobile device under test is Samsung Galaxy S22+ with a Qualcomm Snapdragon SM8450 chip, featuring eight Cortex cores and an Adreno 730 GPU.

**Server:** The server for pose estimation and model training is a machine equipped with an Intel i7-11700 CPU, 32GB memory, and an NVIDIA GeForce RTX 3060 GPU.

**Networking:** We connect the server to a Cisco DPC3941T WiFi router with an Ethernet cable. The mobile device communicates with the server wirelessly over WiFi, which provides 100+ Mbps of bandwidth and ~3 ms round-trip delay.

**Visual Features:** We focus on the well-known SIFT [65] used in SfM and experiment with ORB [84] and SURF [23] (128-dimension version) to demonstrate the generalizability of PIPE. Our method also supports machine-learned features such as SuperPoint [37].

**Metrics:** We leverage the pLEAK metric proposed in §4.4 to evaluate the effectiveness of privacy preservation. Moreover, to assess the accuracy of pose estimation, we utilize the metrics introduced in §2.2, which report position and orientation errors and the recall rates of errors within high, medium, and low-precision intervals.

**Datasets:** We conduct experiments with four well-established datasets for a comprehensive evaluation across various environments.: the outdoor Aachen [88] and GreatCourt [58], and the indoor 7Scenes [92] and NYU depth [72] datasets.

**Image Reconstruction Model:** We consider two state-of-the-art attack models, SIFT-Reconstruction [104] and NinjaDesc [73], which utilize CNN to reconstruct images from extracted features.

### 6.2 Privacy Protection

In this section, we evaluate PIPE's efficacy in protecting user privacy. We first show that PIPE hinders accurate image reconstructions and restricts object detection within the reconstructed images. Subsequently, we employ pLEAK to show PIPE's capability in reducing privacy leakage, which outperforms state-of-the-art methods.

**Evaluation with SSIM and Object Detection.** Figure 6 plots the SSIM and the normalized number of objects detected in images reconstructed using $p$% of SIFT or ORB features selected by PIPE,

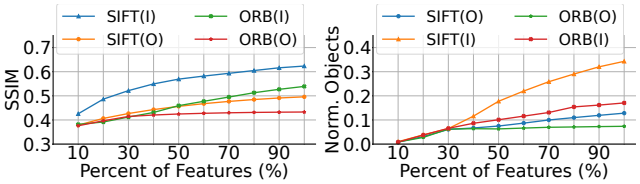



**Figure 6: SSIM (left) and the normalized number of objects (right) of outdoor (O) and indoor (I) images reconstructed using SIFT-Reconstruction [104]. The percent of features $p$ varies from 10 to 100 in steps of 10.**

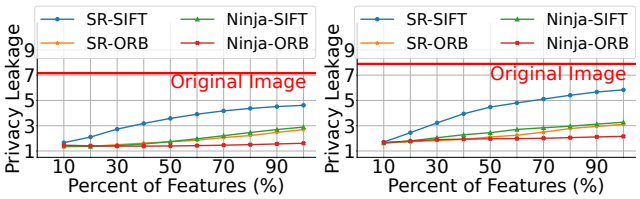



**Figure 7: Privacy leakage of outdoor (left) and indoor (right) images reconstructed using PIPE selected features ($p$ varies from 10 to 100 in steps of 10). The red line indicates privacy leakage of original images.**

with $p$ ranging from 10 to 100 in steps of 10. SURF shows comparable performance with SIFT, probably because both of them use 128-dimensional descriptors, preserving feature distinctiveness and sufficient information for reconstruction. The number of objects is normalized based on the count from the (unprotected) original images, which ranges from 1 to 60 for outdoor and 2 to 69 for indoor scenes. Due to space constraints, we present results from the SIFT-Reconstruction model [104], which outperforms NinjaDesc [73] (*e.g.,* SSIM of 0.50 vs. 0.44 for outdoor images reconstructed from all features). In the following part on evaluating privacy leakage levels, we detail the differences between the two models. As PIPE reduces the uploaded features, there is a marked decrease in SSIM and the number of detected objects. These trends show the efficacy of PIPE in deterring privacy attacks compared with uploading all features for pose estimation.

By selectively uploading features, PIPE reduces the risk of reconstructing images that closely resemble the originals. For example, the SSIM of images reconstructed from using 10% features is lower than 0.43 for outdoor and indoor scenes, indicating a bad similarity to the original images [34]. For comparison, the SSIM of images reconstructed from all SIFT features is 0.50 and 0.63 for outdoor and indoor scenes, respectively, showing a closer resemblance to the originals, which potentially leaks more privacy. While the difference in SSIM for outdoor scenes is 0.07, we will show next this still indicates a noticeable change in privacy risks (*i.e.,* 1.6 vs. 4.6 for privacy-leakage level).

Images reconstructed from reduced features appear blurred, making it challenging for machine learning techniques, such as object detection, to identify private objects. Specifically, indoor images reconstructed using all SIFT features reveal around 38% of objects found in the original images, higher than the 13% observed for outdoor images. This may be attributed to the higher density of objects in indoor environments, resulting in more areas containing

| Method | Aachen (SIFT) | | | Aachen (SURF) | | | Aachen (ORB) | | | 7Scenes (SIFT) | | | 7Scenes (SURF) | | | 7Scenes (ORB) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | High | Med. | Low | High | Med. | Low | High | Med. | Low | High | Med. | Low | High | Med. | Low | High | Med. | Low |
| All | **83.4** | **91.3** | **96.0** | **78.5** | **86.8** | **93.1** | **58.4** | **69.8** | **80.0** | **76.5** | **93.2** | **97.3** | **74.5** | **92.7** | **97.0** | **64.7** | **86.6** | **93.2** |
| *Random* | 68.5 | 77.8 | 85.5 | 62.3 | 72.8 | 82.2 | 35.4 | 47.1 | 59.2 | 60.3 | 81.4 | 88.8 | 58.2 | 80.1 | 87.9 | 47.9 | 71.3 | 81.2 |
| *Score* | 53.8 | 64.1 | 73.1 | 62.1 | 73.2 | 81.7 | 26.3 | 37.3 | 48.7 | 54.3 | 76.8 | 86.4 | 56.9 | 79.7 | 87.3 | 30.8 | 55.3 | 71.0 |
| *Ranked* | 78.9 | **88.8** | **94.9** | 74.0 | 82.3 | 89.9 | 44.1 | 55.1 | 67.5 | 65.3 | 86.8 | 93.8 | 63.1 | 85.2 | 92.1 | 51.4 | 75.7 | 85.4 |
| PIPE | **80.7** | 88.6 | **94.9** | **74.7** | **82.5** | **90.1** | **45.3** | **57.0** | **69.8** | **68.7** | **89.4** | **94.7** | **66.8** | **88.2** | **93.9** | **54.4** | **77.7** | **86.4** |

**Table 5: Pose estimation accuracy on the Aachen and 7Scenes datasets with SIFT and ORB features for five different schemes. Red indicates the best result, and Blue indicates the second-best result.**

identifiable objects being reconstructed. Nevertheless, with PIPE's protection, the proportion of detected objects sharply decreases, dropping to below 1% for both outdoor and indoor images when selecting 10% features.

**Evaluation of Privacy Leakage Level.** We use the pLEAK metric proposed in §4.4 to evaluate the effectiveness of privacy protection. In Figure 7, we plot the privacy-leakage level of images reconstructed with $p$% of features selected by PIPE. We vary $p$ from 10 to 100 in steps of 10. The horizontal red line indicates the privacy-leakage level of original images without protection. We refer to the attack model that reconstructs images from SIFT and ORB features with the SIFT-Reconstruction [104] model as SR-SIFT and SR-ORB and refer to those with NinjaDesc [73] as Ninja-SIFT and Ninja-ORB, respectively. As shown in Figure 7, indoor and outdoor images without protection have a high privacy-leakage level of 7+. In particular, indoor images tend to leak more privacy because they often contain more private objects, for example, in bedrooms.

The SIFT-Reconstruction model, in comparison to the NinjaDesc model, leverages adversarial learning and reconstructs images from coarse to fine granularity, thereby resulting in higher-quality images with a better reconstruction of private content. Consequently, we focus on assessing privacy leakage in images reconstructed with the SIFT-Reconstruction model. As shown in Figure 7, reconstructing images from fewer features selected by PIPE notably reduces privacy leakage, thereby offering improved privacy protection. The privacy-leakage level is low (1.6 and 1.7 for outdoor and indoor images, respectively) when we set $p$ as 10. On the other hand, even when an image is protected by uploading all its extracted features for pose estimation, the level of privacy leakage remains substantial (4.6 and 5.8 for outdoor and indoor images, respectively).

ORB features lead to less privacy leakage than SIFT features for both outdoor and indoor datasets. The difference can be primarily attributed to the distinct characteristics of SIFT and ORB descriptors. SIFT generates a 128-dimensional vector for each feature, with each dimension represented by 8 bits, while ORB utilizes a more compact 256-bit binary string as its descriptor. Thus, ORB features potentially encode less data about images, presenting a lower risk of privacy leakage. However, we show that ORB features perform worse regarding pose estimation accuracy (§6.3).

**Comparison with Prior Work.** We use the pLEAK metric to compare PIPE with state-of-the-art methods that obscure either feature positions [96] or descriptors [40, 73] using the Aachen dataset. We employ the SIFT-Reconstruction model for image reconstruction. For methods that conceal feature positions, we adopt the attack strategy proposed by Speciale *et al.* [96]. For methods focusing on protecting descriptors [40, 73], we provide the model with only feature positions, assuming the descriptors are well-protected.

Protecting positions (descriptors) leads to a privacy-leakage level of 2.8 (2.9), which is less effective compared with the using 10% features (1.6) and 20% features (2.1) provided by PIPE. This is because, compared with merely protecting positions or descriptors, PIPE leaks less privacy by sending only a limited number of features from the client to the server.

## 6.3 Pose Estimation Accuracy

We evaluate the pose estimation accuracy with four different feature selection methods, namely *Random*, *Score*, *Ranked*, and PIPE. *Random* selects features randomly. *Score* selects features based on their score values, which are generated by feature extraction algorithms and indicate the distinctiveness and reliability of features [65, 84]. To analyze the impact of individual design components of PIPE, we conduct an ablation study where *Ranked* selects features solely based on their ranking (§4.2), while PIPE incorporates both ranking (§4.2) and spatial distribution (§4.3) for feature selection. We compare these methods with a baseline that utilizes all features for pose estimation (without privacy protection).

Table 5 compares the baseline and different feature selection methods on the outdoor Aachen and indoor 7Scenes datasets with $p$ set as 10. ORB performs the worst because it is less scale-invariant than SIFT and SURF [98], which affects the effectiveness of feature matching and subsequent pose estimation. Compared with SIFT, although SURF is also scale-invariant, it uses an approximate filtering approach for faster feature extraction, which may detect keypoints with lower distinctiveness, reducing the pose estimation accuracy. Since pose estimation with SIFT features performs better than ORB and SURF features, we mainly focus on pose estimation with SIFT features in the following analysis, though the results with ORB and SURF features show similar trends.

Our results highlight the effectiveness of PIPE, which offers reliable pose estimation by considering both the ranking and spatial distribution of features. On the Aachen dataset with SIFT features, *Score* is the least effective, indicating that the score values generated by feature extraction algorithms do not offer valuable hints on whether a feature is an inlier. Compared with *Random*, *Ranked* improves the recall rate under the high-precision interval from 68.5% to 78.9%, as it selects more inliers (118) than *Random* (38). Furthermore, PIPE selects high-ranking and scattered features, leading to more accurate pose estimation comparable to the performance

| Resolution | Recall @ Thresh (%) | | |
|---|---|---|---|
| | High | Med. | Low |
| 1600×1200 | 88.0 | 93.1 | 96.7 |
| 1024×768 | 87.9 | 93.4 | 96.6 |
| 768×576 | 80.6 | 88.3 | 92.8 |
| 512×384 | 74.0 | 83.0 | 87.7 |

**Table 6: Pose estimation accuracy of PIPE on the Aachen dataset with varied resolutions.**

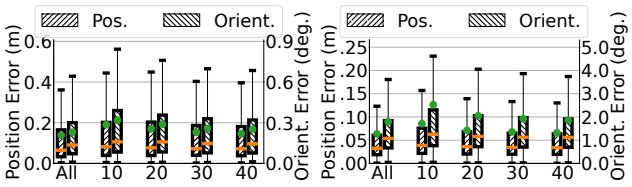



**Figure 7: Pose estimation errors on Aachen (left) and 7Scenes (right) with the SIFT features selected by PIPE compared with using all features. $p$ varies from 10 to 40 at steps of 10.**

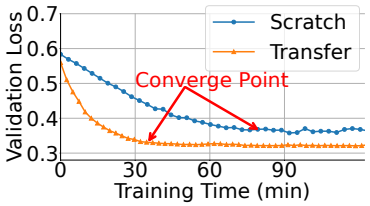



**Figure 8: Comparison of validation loss between *Scratch* and *Transfer* models on *MapB* of Aachen.**

| Method | Aachen | | | 7Scenes | | |
|---|---|---|---|---|---|---|
| | High | Med. | Low | High | Med. | Low |
| Full Training | **77.1** | **86.3** | **92.1** | **68.1** | **88.7** | **94.2** |
| *Random* | 67.2 | 76.9 | 84.2 | 59.4 | 81.1 | 88.2 |
| Pre-trained on *MapA* | 72.7 | 81.9 | 88.7 | 61.9 | 83.4 | 90.6 |
| Scratch (30 mins) | 72.2 | 81.1 | 88.5 | 64.3 | 86.1 | 92.9 |
| Transfer (30 mins) | **76.9** | **85.8** | **91.4** | **68.0** | **88.4** | **93.7** |

**Table 7: Localization results on *MapB* of Aachen and 7Scenes datasets with transfer learning. Red indicates the best result, and Blue indicates the second-best result.**

achieved in prior work [40, 73, 96] for privacy-preserving image-based pose estimation. The performance of PIPE is close to the baseline, falling short by only 2.7% under the high-precision interval. The 75th percentile of pose estimation errors for PIPE is relatively low, with a position error of ~0.20m and an orientation error of ~0.39°. For comparison, the 75th percentile of pose estimation errors of the baseline is around 0.17m and 0.31°.

On the 7Scenes dataset, the gap in recall rates under the high-precision interval between PIPE and the baseline is 7.8%, which is larger than that on the Aachen dataset (2.7%). This is mainly due to the high rotation errors of the baseline. The 75th percentile of rotation errors for the baseline is 1.9°, which is close to the threshold of the high-precision interval (2°). Thus, a slight increase in rotation errors leads to a considerable drop in recall rate under the high-precision interval. Nevertheless, the 75th percentile of pose estimation errors for PIPE (0.07m and 2.3°) is close to the baseline (0.06m and 1.9°). Compared with *Random*, PIPE improves pose estimation accuracy under high-level precision by 8.4%. Moreover, PIPE achieves recall rates similar to the baseline under medium and low-precision intervals. Our observations from the GreatCourt and NYU depth datasets are consistent with the findings on the Aachen and 7Scenes datasets.

Table 6 shows the pose estimation accuracy of PIPE on the Aachen dataset with localization images at resolutions of 1600×1200, 1024×768, 768×576, and 512×384. Similar to our findings in §4.1, reducing the resolution from 1600×1200 to 1024×768 has limited impacts on pose estimation accuracy (88.0% vs. 87.9% for high-precision), while further reductions lead to lower accuracy (<80.5%).

To further explore how the pose estimation accuracy varies among different levels of privacy protections, we conduct experiments by varying $p$ from 10 to 40 at the step of 10 to show the pose estimation accuracy. Figure 7 shows the pose estimation errors for the Aachen and 7Scenes datasets. In these plots, we depict the 25th and 75th percentiles, medium, mean (green dots), and lower and upper whiskers [36]. As shown in Figure 7, pose estimation errors decrease as we include more features, consequently incorporating more inliers for pose estimation. While PIPE can deliver reasonably accurate pose estimation when using only 10% features, users who are willing to accept lower-level protection can benefit from improved pose estimation.

## 6.4 Transferability of Ranking Model

We further show that the feature-ranking model is transferable across different maps, enabling rapid training that facilitates the application of PIPE across various maps to cover wide areas. In our experiment, we partition the Aachen map into two parts, *MapA* and *MapB*. This segmentation utilizes the spatial coordinates of map images and horizontally bisects the map at its median coordinate. Certain areas from *MapA* may be visible in images from *MapB*, and vice versa. To ensure the robustness of the experimental design, we judiciously remove areas that are discernible from both *MapA* and *MapB*. We also evaluate the 7Scenes dataset, using similar methods to separate the map into *MapA* and *MapB*.

We train two feature-ranking models with 80% of randomly selected map features from *MapB*, with the remaining 20% for validation. The first model, referred to as *Scratch*, trains with only map features from *MapB*. The second model, referred to as *Transfer*, transfers a pre-trained model, which is trained with all map features from *MapA*, and fine-tunes it with map features from *MapB*. Figure 8 shows that the *Transfer* model reaches convergence in around 35 minutes, reducing the training time by more than 56% compared with *Scratch*, which requires over 80 minutes.

Furthermore, we evaluate the pose estimation accuracy on *MapB* of Aachen and 7Scenes datasets with four ranking models. We train the baseline model using the map images from *MapB* over the full epochs. The other three models include a pre-trained model using

| Stage | Feature Extraction | Feature Ranking | Feature Selection | Transmission | Matching & P3P RANSAC on Server | E2E |
|---|---|---|---|---|---|---|
| Baseline | 137±5.5 | N/A | N/A | 28±5 | 937±186 | ~1,102 |
| 10% *Random* | 219±16 | N/A | <1 | 4.9±0.6 | 490±124 | ~749 |
| 10% PIPE (SIFT) | | 125±38 | 14±11 | | | ~853 |
| 20% PIPE (SIFT) | | | | 6.4±1.1 | 608±142 | ~973 |
| 40% PIPE (SIFT) | | | | 7.9±1.6 | 729±160 | ~1,095 |
| 10% PIPE (SURF) | 73±12 | 127±36 | 15±10 | 5.1±0.6 | 501±127 | ~721 |
| 10% PIPE (ORB) | 49±6 | 67±17 | 11±8 | 2.1±0.4 | 224±79 | ~354 |

**Table 8: Latency (ms) comparison of PIPE and the baseline system.**

map images from *MapA*, a model that we train from scratch on *MapB* for 30 minutes, and a model that is transferred from the pre-trained model to *MapB* with 30 minutes of training. We also evaluate the *Random* method for comparison.

Table 7 shows that the feature-ranking model is transferable across different maps, enabling rapid training and effective adaptation to new environments. Compared with *Random*, the pre-trained model from *MapA* marginally increases the pose estimation accuracy, with a 4.5% and 2.5% increase on the Aachen and 7Scenes datasets, respectively, under the high-precision interval. This improvement is attributed to the shared structural and characteristic similarities in the features from both maps. Furthermore, transferring the pre-trained model on *MapB* for 30 minutes enables the model to acquire new characteristics from *MapB* features. Consequently, this model's pose estimation accuracy is comparable to the model trained over the full epochs (>200 minutes). On the other hand, compared with the baseline, the model that trains from scratch for 30 minutes shows a 4.9% and 3.8% decrease in the two datasets, respectively, under the high-precision interval.

## 6.5 End-to-end Latency

We measure the end-to-end latency of PIPE by breaking it into client-side feature preparation, data transfer, and server-side pose estimation. We show the results in Table 8. Using the Aachen dataset, we evaluate PIPE with $p$ set to 10, 20, and 40 for different levels of protection (*i.e.,* satisfying almost all, 75% and 50% users).

PIPE's client-side latency is 358±66 ms, with 219±16 ms for feature extraction, 125±38 ms for ranking, and 14±11 ms for selection. The network latency is around 4.9±0.6 ms ($p$=10), 6.4±1.1 ms ($p$=20), and 7.9±1.6 ms ($p$=40). The server-side latency, which consists of feature matching and P$n$P RANSAC, takes on average 490±124 ms ($p$=10), 608±142 ms ($p$=20), and 729±160 ms ($p$=40). Thus, the average end-to-end latency of PIPE is around 853 ms ($p$=10), 973 ms ($p$=20), and 1,095 ms ($p$=40). Compared with *Random*, PIPE significantly boosts pose estimation accuracy with an extra latency of only around 104 ms for on-device feature ranking and selection. As a result, we believe the latency is well-suited for practical use, as only keyframes are utilized for server-based localization, and the process is primarily for map alignment rather than continuous real-time tracking (§3).

We also compare PIPE with the baseline system that sends images for pose estimation without privacy protection. The images' resolutions are the same for an apple-to-apple comparison. Due to more features to match and for P$n$P RANSAC, the baseline's computational load increases significantly, resulting in an average end-to-end latency of around 1,102 ms. In comparison, PIPE reduces end-to-end latency across all protection levels, saving around 22.6%, 11.7%, and 0.6% for high, medium, and low-level protections, respectively. Furthermore, compared with leveraging SURF and ORB features for pose estimation, SIFT offers higher accuracy at the expense of increased computation latency. However, as discussed in §3, PIPE focuses on the localization of keyframe images, where accurate pose estimation is the most important, making SIFT the preferred choice.

## 6.6 Energy Consumption

To profile the energy consumption of PIPE, we execute it on Samsung Galaxy S22+ for 2 hours. We compare PIPE with two baseline systems that upload either captured images or all of the extracted features from them to a server for pose estimation. We start each experiment on the fully-charged device. After the 2-hour experiment, the battery level decreases from 100% to 78% for uploading images, to 72% for extracting and uploading features, and to 68% for PIPE. The differences in energy consumption come from the locally performed feature extraction, ranking, and selection in PIPE. Overall, we believe the energy consumption of PIPE is acceptable.

# 7 Discussion

**Privacy Leakage *vs.* Information Leakage.** Addressing privacy leakage in the context of image-based pose estimation is a distinct challenge, due to the subjective nature of privacy [21, 57, 107]. However, privacy leakage is closely related to the broader and more objective concept of information leakage. Information leakage involves unauthorized access or dissemination of potentially sensitive or valuable data, whether or not it is personally identifiable, and is inherently more measurable and quantifiable. Privacy leakage, in contrast, specifically pertains to personal data, which can vary greatly among individuals, making it harder to define and quantify. Because of this relationship, privacy leakage is a subset of information leakage. Hence, by addressing the wider issues of information leakage, we can indirectly mitigate the risks of privacy breaches.

**Fully Homomorphic Encryption (FHE) & Multi-party Computation (MPC)** are powerful cryptography techniques, but they may not be suitable for pose estimation. FHE [46] allows computations to be carried out on encrypted data without decrypting it [46]. However, FHE struggles to solve the non-polynomial operations P$n$P [105], and replacing P$n$P with direct linear transformation

(DLT) [52], which simplifies pose estimation by linearizing it, results in degraded pose estimation accuracy [105]. Additionally, FHE dramatically increases computation overhead [14, 71] and data size after encryption (*e.g.,* >3000× [105]), making it impractical for real-time applications. Similarly, MPC [49], which allows multiple parties to jointly compute a function while keeping their inputs private, incurs high communication overhead. For instance, data transmission for a single image localization can exceed 64GB [32], leading to substantial delay.

**Machine-learned Features.** To validate the generalizability of PIPE, we explore its effectiveness with machine-learned features such as Superpoint [37] on the Aachen dataset, setting $p$ to 10. The number of Superpoint features is comparable with SIFT. Our results indicate PIPE still performs better than *Random*. For the high-precision interval, *Random* and PIPE result in a 6.2% and 3% recall drop, respectively. This comparative analysis further affirms the generalizability and efficacy of PIPE. Note that while Superpoint provides more accurate pose estimation than SIFT and ORB with its superior matching ability [37, 85], the high computational overhead associated with Superpoint makes it unsuitable for real-time pose estimation. Moreover, Superpoint leaks more privacy than SIFT, probably due to its higher feature dimensions (256 in Suerpoint vs. 128 in SIFT). For example, for outdoor scenes, the average privacy-leakage level from images reconstructed from 10% of SIFT features is only 1.6, while a similar number of Superpoint features results in a leakage level of 2.5, equivalent to using around 30% of SIFT features for image reconstruction.

**Privacy-leakage Metric.** In our work, we propose a new metric, pLEAK, as existing metrics are not directly applicable in image-based 6DoF pose estimation for mobile devices. Differential privacy [41] primarily focuses on statistical data and may not be able to directly resolve privacy concerns in visual data targeted by PIPE. Specifically, when privacy leakage arises from identifiable visual features utilized for pose estimation, traditional noise addition techniques may fall short, resulting in poor pose estimation accuracy [79]. The goal of pLEAK is to provide a practical metric closely aligned with human perceptions of privacy. As such, pLEAK is not a foundational guarantee of privacy preservation but a measure of *perceived* privacy leakage. While its effectiveness may vary with the development of more advanced algorithms, the same technology can be repurposed to maintain utility.

**Improving Feature Ranking.** To improve the generalizability of feature ranking, incorporating a complex CNN model that takes full image content instead of features' descriptors as input can capture broader contextual information. This approach allows the model to adapt to diverse environments more effectively. However, the increased model complexity may introduce significant computational overhead, including higher inference latency and greater energy consumption, making it unsuitable for mobile devices. To address these challenges, future work may explore model distillation to enhance generalization while maintaining efficiency. In this framework, a high-capacity *teacher* network is first trained to learn a robust and adaptable feature ranking strategy across different scenes. A smaller *student* model is then optimized to distill the *teacher* network's ranking capability on a specific map to reduce computational demands substantially. This method can potentially enable feature ranking models to maintain adaptability across varying environments while remaining efficient for real-time processing on mobile devices.

## 8 Related Work

**Privacy-preserving Vision Techniques.** To address privacy concerns from users, various schemes have been proposed to protect sensitive visual information [40, 48, 82, 90, 95, 96]. For example, Dusmanu *et al.* [40] propose to embed feature descriptors to affine subspace to conceal private content. Another line of work obfuscates feature positions by representing feature points with lines in 2D [96] and 3D spaces [48, 90, 95]. Instead of concealing feature positions or descriptors, PIPE reduces the number of uploaded features to mitigate privacy leakage in localization images.

**Privacy-preserving Applications.** Privacy preservation has been widely studied for various applications [15, 27, 30, 31, 33, 62, 80, 103, 109, 110]. For example, YANA [62] protects each user's private interests from the recommendation server by grouping users with diverse interests. MetaFL [31] leverages federated learning to preserve privacy for authentication in virtual reality. I-Pic [15] utilizes privacy defined by nearby users to edit captured images for privacy preservation. Previous studies employ securely reversible transformed images [103], differential privacy [27], and trusted execution environments (TEE) [80] for video analytics. PIPE targets a different application and protects privacy in the images that are sent to the server for localization.

**Privacy-preserving Localization.** Preserving privacy has been a key research topic in different types of localization techniques, especially for fingerprint-based WiFi localization [22]. Commonly used methods include efficient location obfuscation algorithms [91] such as $k$-anonymity [61, 111], homomorphic encryption [19, 63, 93, 108], secure two-party computation [56], and antenna pattern synthesis [101]. Different from the above work, image-based pose estimation systems may leak more sensitive (environmental) information captured in images than WiFi-based schemes.

## 9 Conclusion

In this paper, we designed, implemented, and evaluated PIPE, a practical system for privacy-preserving image-based 6DoF pose estimation of mobile devices. Building on the key insight that selective upload of a small fraction of features could protect private information in localization images, we addressed various challenges such as computation-intensive on-device feature extraction, feature ranking with limited knowledge, reduced pose estimation accuracy with clustered high-ranking features, and effective quantification of privacy leakage in images. Our extensive performance evaluation showed that PIPE effectively preserves privacy with limited impact on pose estimation accuracy and reduces end-to-end latency by up to 22.6% compared with the baseline.

## Acknowledgment

We thank the anonymous reviewers and our shepherd for their insightful feedback, as well as the participants of our user studies for their valuable contributions. This work was partially supported by the National Science Foundation under Grant CNS-2235049 and a research award from Meta.

## References


[1] 2017. Who Is Thinking About Security and Privacy for Augmented Reality? https://www.technologyreview.com/2017/10/19/105305/who-is-thinking-about-security-and-privacy-for-augmented-reality/. [accessed on 03/01/2025].

[2] 2018. Privacy Manifesto for AR Cloud Solutions. https://medium.com/openarcloud/privacy-manifesto-for-ar-cloud-solutions-9507543f50b6. [accessed on 03/01/2025].

[3] 2019. Using Global Localization to Improve Navigation. https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html. [accessed on 03/01/2025].

[4] 2022. ARCore. https://developers.google.com/ar. [accessed on 03/01/2025].

[5] 2022. Build Global-Scale, Immersive, Location-based AR Experiences with the ARCore Geospatial API. https://developers.google.com/ar/develop/geospatial. [accessed on 03/01/2025].

[6] 2023. ARKit. https://developer.apple.com/documentation/arkit. [accessed on 03/01/2025].

[7] 2023. Augmented Reality. https://en.wikipedia.org/wiki/Augmented_reality#Uses. [accessed on 03/01/2025].

[8] 2023. Digital and Reality AR ready to merge. https://immersal.com/.

[9] 2023. Lasso (statistics). https://en.wikipedia.org/wiki/Lasso_(statistics). [accessed on 03/01/2025].

[10] 2023. Multilayer Perceptron. https://en.wikipedia.org/wiki/Multilayer_perceptron. [accessed on 03/01/2025].

[11] 2023. Root-mean-square Deviation. https://en.wikipedia.org/wiki/Root-mean-square_deviation. [accessed on 03/01/2025].

[12] 2023. t-distributed Stochastic Neighbor Embedding. https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding. [accessed on 03/01/2025].

[13] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/

[14] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Computing Surveys (Csur)* 51, 4 (2018), 1–35.

[15] Paarijaat Aditya, Rijurekha Sen, Peter Druschel, Seong Joon Oh, Rodrigo Benenson, Mario Fritz, Bernt Schiele, Bobby Bhattacharjee, and Tong Tong Wu. 2016. I-Pic: A Platform for Privacy-Compliant Image Capture. In *Proceedings of ACM MobiSys*. https://doi.org/10.1145/2906388.2906412

[16] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. 2009. Building Rome in a Day. In *Proceedings of International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2009.5459148

[17] Berker Ağır, Kévin Huguenin, Urs Hengartner, and Jean-Pierre Hubaux. 2016. On the Privacy Implications of Location Semantics. In *Proceedings of International Symposium on Privacy Enhancing Technologies Symposium (PETS)*. https://doi.org/10.1515/popets-2016-0034

[18] Fawad Ahmad, Hang Qiu, Ray Eells, Fan Bai, and Ramesh Govindan. 2020. CarMap: Fast 3D Feature Map Updates for Automobiles. In *Proceedings of USENIX NSDI*. https://www.usenix.org/conference/nsdi20/presentation/ahmad

[19] Amr Alanwar, Yasser Shoukry, Supriyo Chakraborty, Paul Martin, Paulo Tabuada, and Mani Srivastava. 2017. PrOLoc: Resilient Localization with Private Observers Using Partial Homomorphic Encryption. In *Proceedings of ACM/IEEE IPSN*. https://doi.org/10.1145/3055031.3055033

[20] AugmentedCity. [n. d.]. Create City-Scale Augmented Reality Experiences, Apps and Products. https://www.augmented.city/. [accessed on 03/01/2025].

[21] Gonul Ayci, Murat Sensoy, Arzucan Özgür, and Pinar Yolum. 2023. Uncertainty-Aware Personal Assistant for Making Personalized Privacy Decisions. *ACM Transactions on Internet Technology* 23, 1 (2023), 1–24. https://doi.org/10.1145/3561820

[22] Paramvir Bahl and Venkata N Padmanabhan. 2000. RADAR: An In-Building RF-based User Location and Tracking System. In *Proceedings of IEEE INFOCOM*. https://doi.org/10.1109/INFCOM.2000.832252

[23] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. SURF: Speeded Up Robust Features. *Lecture Notes in Computer Science* 3951 (2006), 404–417. https://doi.org/10.1007/11744023_32

[24] Ali J Ben Ali, Marziye Kouroshli, Sofiya Semenova, Zakieh Sadat Hashemifar, Steven Y Ko, and Karthik Dantu. 2022. Edge-SLAM: Edge-assisted Visual Simultaneous Localization and Mapping. *ACM Transactions on Embedded Computing Systems* 22, 1 (2022), 1–31.

[25] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal Speed and Accuracy of Object Detection. https://arxiv.org/abs/2004.10934. [accessed on 03/01/2025].

[26] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[27] Frank Cangialosi, Neil Agarwal, Venkat Arun, Junchen Jiang, Srinivas Narayana, Anand Sarwate, and Ravi Netravali. 2022. Privid: Practical, Privacy-Preserving Video Analytics Queries. In *Proceedings of USENIX NSDI*. https://www.usenix.org/conference/nsdi22/presentation/cangialosi

[28] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. 2021. How Privacy-Preserving Are Line Clouds? Recovering Scene Details From 3D Lines. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR46437.2021.01541

[29] Kaifei Chen, Tong Li, Hyung-Sin Kim, David E Culler, and Randy H Katz. 2018. MARVEL: Enabling Mobile Augmented Reality with Low Energy and Low Latency. In *Proceedings of ACM SenSys*. https://doi.org/10.1145/3274783.3274834

[30] Ruizhi Cheng, Songqing Chen, and Bo Han. 2023. Toward Zero-trust Security for the Metaverse. *IEEE Communications Magazine* 62, 2 (2023), 156–162.

[31] Ruizhi Cheng, Yuetong Wu, Ashish Kundu, Hugo Latapie, Myungjin Lee, Songqing Chen, and Bo Han. 2024. MetaFL: Privacy-preserving User Authentication in Virtual Reality with Federated Learning. In *Proceedings of ACM SenSys*.

[32] James Choncholas, Pujith Kachana, André Mateus, Gregoire Phillips, and Ada Gavrilovska. 2024. Snail: Secure Single Iteration Localization. In *Proceedings of Privacy Enhancing Technologies Symposium*.

[33] Matthew Corbett, Brendan David-John, Jiacheng Shang, Y Charlie Hu, and Bo Ji. 2023. BystandAR: Protecting Bystander Visual Data in Augmented Reality Systems. In *Proceedings of MobiSys*. https://doi.org/10.1145/3581791.3596830

[34] Eduardo Cuervo, Alec Wolman, Landon P. Cox, Kiron Lebeck, Ali Razeen, Stefan Saroiu, and Madanlal Musuvathi. 2015. Kahawai: High-Quality Mobile Gaming Using GPU Offload. In *Proceedings of ACM MobiSys*. https://doi.org/10.1145/2742647.2742657

[35] 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. 2021. Mitigating Reverse Engineering Attacks on Local Feature Descriptors. In *Proceedings of British Machine Vision Conference (BMVC)*.

[36] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. 2005. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer.

[37] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. https://doi.org/10.1109/CVPRW.2018.00060

[38] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. 2022. Learning To Detect Scene Landmarks for Camera Localization. In *Proceedings of the IEEE/CVF CVPR*.

[39] Alexey Dosovitskiy and Thomas Brox. 2016. Inverting Visual Representations with Convolutional Networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2016.522

[40] Mihai Dusmanu, Johannes L. Schönberger, Sudipta N. Sinha, and Marc Pollefeys. 2021. Privacy-Preserving Image Features via Adversarial Affine Subspace Embeddings. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR46437.2021.01404

[41] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of Theory of Cryptography Conference (TCC)*. https://doi.org/10.1007/11681878_14

[42] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. 1997. The Farthest Point Strategy for Progressive Image Sampling. *IEEE Transactions on Image Processing* 6, 9 (1997), 1305–1315. https://doi.org/10.1109/83.623193

[43] Xuewei Feng, Qi Li, Kun Sun, Yuxiang Yang, and Ke Xu. 2023. Man-in-the-Middle Attacks without Rogue AP: When WPAs Meet ICMP Redirects. In *Proceedings of IEEE Symposium on Security and Privacy (SP)*. https://doi.org/10.1109/SP46215.2023.10179441

[44] Martin A. Fischler and Robert C. Bolles. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24, 6 (1981), 381–395. https://doi.org/10.1145/358669.358692

[45] Pascal Fua and Vincent Lepetit. 2007. Vision Based 3D Tracking and Pose Estimation for Mixed Reality. In *Emerging Technologies of Augmented Reality: Interfaces and Design*. IGI Global, Hershey, 1–22. https://doi.org/10.4018/978-1-59904-066-0.ch001

[46] Craig Gentry. 2009. *A Fully Homomorphic Encryption Scheme*. Ph. D. Dissertation.

[47] Marcel Geppert, Viktor Larsson, Johannes L Schönberger, and Marc Pollefeys. 2022. Privacy Preserving Partial Localization. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR52688.2022.01682

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

*of European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-030-58452-8_20

[49] Oded Goldreich. 1998. Secure multi-party computation. *Manuscript. Preliminary version* 78, 110 (1998), 1–108.

[50] Oded Goldreich. 2004. *Foundations of Cryptography*. Cambridge University Press Cambridge.

[51] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. 2013. Rotation Averaging. *International Journal of Computer Vision* 103 (2013), 267–305. https://doi.org/10.1007/s11263-012-0601-0

[52] Richard Hartley and Andrew Zisserman. 2004. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[53] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

[54] Tianyi Hu, Tim Scargill, Fan Yang, Ying Chen, Guohao Lan, and Maria Gorlatova. 2024. SEESys: Online Pose Error Estimation System for Visual SLAM. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*.

[55] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. 2020. Robust Image Retrieval-based Visual Localization using Kapture. https://arxiv.org/pdf/2007.13867. [accessed on 03/01/2025].

[56] Kimmo Järvinen, Helena Leppäkoski, Elena-Simona Lohan, Philipp Richter, Thomas Schneider, Oleksandr Tkachenko, and Zheng Yang. 2019. PILOT: Practical Privacy-preserving Indoor Localization using Outsourcing. In *Proceedings of IEEE European Symposium on Security and Privacy (EuroS&P)*. https://doi.org/10.1109/EuroSP.2019.00040

[57] Rui Jiao, Lan Zhang, and Anran Li. 2020. Ieye: Personalized Image Privacy Detection. In *Proceedings of International Conference on Big Data Computing and Communications (BIGCOM)*. https://doi.org/10.1109/BigCom51056.2020.00020

[58] Alex Kendall, Matthew Grimes, and Roberto Cipolla. 2015. Posenet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2015.336

[59] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980. [accessed on 03/01/2025].

[60] Georg Klein and David Murray. 2009. Parallel Tracking and Mapping on a Camera Phone. In *Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR)*. https://doi.org/10.1109/ISMAR.2009.5336495

[61] Andreas Konstantinidis, Georgios Chatzimilioudis, Demetrios Zeinalipour-Yazti, Paschalis Mpeis, Nikos Pelekis, and Yannis Theodoridis. 2015. Privacy-Preserving Indoor Localization on Smartphones. *IEEE Transactions on Knowledge and Data Engineering* 27, 11 (2015), 3042–3055. https://doi.org/10.1109/TKDE.2015.2441724

[62] Dongsheng Li, Qin Lv, Li Shang, and Ning Gu. 2017. Efficient Privacy-Preserving Content Recommendation for Online Social Communities. *Neurocomputing* 219, C (2017), 440–454. https://doi.org/10.1016/j.neucom.2016.09.059

[63] Hong Li, Limin Sun, Haojin Zhu, Xiang Lu, and Xiuzhen Cheng. 2014. Achieving Privacy Preservation in WiFi Fingerprint-Based Localization. In *Proceedings of IEEE INFOCOM*. https://doi.org/10.1109/INFOCOM.2014.6848178

[64] Shuangfeng Li. 2020. Tensorflow Lite: On-device Machine Learning Framework. *Journal of Computer Research and Development* 57 (2020), 1839. http://dx.doi.org/10.7544/issn1000-1239.2020.20200291

[65] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Journal of Computer Vision* 60, 2 (2004), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[66] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding Deep Image Representations by Inverting Them. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2015.7299155

[67] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. 2016. Pose Estimation for Augmented Reality: A Hands-On Survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 22, 12 (2016), 2633–2651. https://doi.org/10.1109/TVCG.2015.2513408

[68] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. 2014. Scalable 6-DOF Localization on Mobile Devices. In *Proceedings of European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-319-10605-2_18

[69] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Ali Jalali, Ahmed Taha Taha Elthakeb, Dean Tullsen, and Hadi Esmaeilzadeh. 2021. Not All Features Are Equal: Discovering Essential Features for Preserving Prediction Privacy. In *Proceedings of the Web Conference*. https://doi.org/10.1145/3442381.3449965

[70] Raul Mur-Artal and Juan D Tardós. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics* 33, 5 (2017), 1255–1262.

[71] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. 2011. Can Homomorphic Encryption be Practical?. In *Proceedings of the 3rd ACM workshop on Cloud Computing Security Workshop*.

[72] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *Proceedings of European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-642-33715-4_54

[73] Tony Ng, Hyo Jin Kim, Vincent T Lee, Daniel DeTone, Tsun-Yi Yang, Tianwei Shen, Eddy Ilg, Vassileios Balntas, Krystian Mikolajczyk, and Chris Sweeney. 2022. NinjaDesc: Content-Concealing Visual Descriptors via Adversarial Learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR52688.2022.01246

[74] Mary Lynne Nielsen. 2015. Augmented Reality and its Impact on the Internet, Security, and Privacy. https://beyondstandards.ieee.org/augmented-reality-and-its-impact-on-the-internet-security-and-privacy/. [accessed on 03/01/2025].

[75] Helen Nissenbaum. 2009. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford University Press.

[76] OpenARCloud. 2021. A reference Open Spatial Computing Platform (OSCP). https://www.openarcloud.org/oscp. [accessed on 03/01/2025].

[77] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[78] Francesco Pittaluga, Sanjeev J. Koppal, Sing Bing Kang, and Sudipta N. Sinha. 2019. Revealing Scenes by Inverting Structure from Motion Reconstructions. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2019.00023

[79] Francesco Pittaluga and Bingbing Zhuang. 2023. LDP-FEAT: Image Features with Local Differential Privacy. In *Proceedings of the IEEE/CVF CVPR*.

[80] Rishabh Poddar, Ganesh Ananthanarayanan, Srinath Setty, Stavros Volos, and Raluca Ada Popa. 2020. Visor: Privacy-Preserving Video Analytics as a Cloud Service. In *Proceedings of USENIX Security*. https://www.usenix.org/conference/usenixsecurity20/presentation/poddar

[81] Prolific. 2024. Easily Find Vetted Research Participants and AI Taskers at Scale. https://www.prolific.com/. [accessed on 03/01/2025].

[82] Zhan Qin, Jingbo Yan, Kui Ren, Chang Wen Chen, and Cong Wang. 2014. Towards Efficient Privacy-preserving Image Feature Extraction in Cloud Computing. In *Proceedings of ACM International Conference on Multimedia (MM)*. https://doi.org/10.1145/2647868.2654941

[83] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. 2019. R2D2: Repeatable and Reliable Detector and Descriptor. In *Proceedings of NeurIPS*. https://dl.acm.org/doi/10.5555/3454287.3455400

[84] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2011.6126544

[85] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 2019. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2019.01300

[86] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 2011. Fast Image-Based Localization using Direct 2D-to-3D Matching. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2011.6126302

[87] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 2016. Efficient & Effective Prioritized Matching for Large-scale Image-based Localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 9 (2016), 1744–1756. https://doi.org/10.1109/TPAMI.2016.2611662

[88] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. 2018. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2018.00897

[89] Johannes L. Schönberger and Jan-Michael Frahm. 2016. Structure-From-Motion Revisited. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2016.445

[90] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. 2020. Privacy Preserving Visual SLAM. In *Proceedings of European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-030-58542-6_7

[91] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2012. Protecting Location Privacy: Optimal Strategy against Localization Attacks. In *Proceedings of ACM Conference on Computer and Communications Security (CCS)*. https://doi.org/10.1145/2382196.2382261

[92] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. 2013. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2013.377

[93] Tao Shu, Yingying Chen, Jie Yang, and Albert Williams. 2014. Multi-lateral Privacy-Preserving Localization in Pervasive Environments. In *Proceedings of IEEE INFOCOM*. https://doi.org/10.1109/INFOCOM.2014.6848176

[94] Daniel J Solove. 2002. Conceptualizing Privacy. *California Law Review* 90, 4 (2002), 1087–1155.

[95] Pablo Speciale, Johannes L. Schönberger, Sing Bing Kang, Sudipta N. Sinha, and Marc Pollefeys. 2019. Privacy Preserving Image-Based Localization. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2019.00564

[96] Pablo Speciale, Johannes L. Schönberger, Sudipta N. Sinha, and Marc Pollefeys. 2019. Privacy Preserving Image Queries for Camera Localization. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2019.00157

[97] Milan Stute, Sashank Narain, Alex Mariotto, Alexander Heinrich, David Kreitschmann, Guevara Noubir, and Matthias Hollick. 2019. A Billion Open Interfaces for Eve and Mallory: MitM, DoS, and Tracking Attacks on iOS and macOS Through Apple Wireless Direct Link. In *Proceedings of USENIX Security*. https://www.usenix.org/conference/usenixsecurity19/presentation/stute

[98] Shaharyar Ahmed Khan Tareen and Zahra Saleem. 2018. A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In *Proceedings of International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. https://doi.org/10.1109/ICOMET.2018.8346440

[99] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. 1999. Bundle Adjustment – A Modern Synthesis. In *Proceedings of International Workshop on Vision Algorithms (IWVA)*. https://doi.org/10.1007/3-540-44480-7_21

[100] Thijs Veugen, Frank Blom, Sebastiaan JA De Hoogh, and Zekeriya Erkin. 2015. Secure Comparison Protocols in the Semi-honest Model. *IEEE Journal of Selected Topics in Signal Processing* 9, 7 (2015), 1217–1228.

[101] Ting Wang and Yaling Yang. 2011. Location Privacy Protection from RSS Localization System Using Antenna Pattern Synthesis. In *Proceedings of IEEE INFOCOM*. https://doi.org/10.1109/INFCOM.2011.5935061

[102] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

[103] Hao Wu, Xuejin Tian, Minghao Li, Yunxin Liu, Ganesh Ananthanarayanan, Fengyuan Xu, and Sheng Zhong. 2021. PECAM: Privacy-enhanced Video Streaming and Analytics via Securely-Reversible Transformation. In *Proceedings of MobiCom*. https://doi.org/10.1145/3447993.3448618

[104] Haiwei Wu and Jiantao Zhou. 2021. Privacy Leakage of SIFT Features via Deep Generative Model Based Image Reconstruction. *IEEE Transactions on Information Forensics and Security* 16 (2021), 2973–2985. https://doi.org/10.1109/TIFS.2021.3070427

[105] Nan Wu, Ruizhi Cheng, Songqing Chen, and Bo Han. 2022. Preserving Privacy in Mobile Spatial Computing. In *Proceedings of NOSSDAV*. https://doi.org/10.1145/3534088.3534343

[106] Yi Wu, Cong Shi, Tianfang Zhang, Payton Walker, Jian Liu, Nitesh Saxena, and Yingying Chen. 2023. Privacy Leakage via Unrestricted Motion-Position Sensors in the Age of Virtual Reality: A Study of Snooping Typed Input on Virtual Keyboards. In *Proceedings of IEEE Symposium on Security and Privacy (S&P)*.

[107] Zhujun Xiao, Jenna Cryan, Yuanshun Yao, Yi Hong Gordon Cheo, Yuanchao Shu, Stefan Saroiu, Ben Y. Zhao, and Haitao Zheng. 2023. "My face, my rules": Enabling Personalized Protection against Unacceptable Face Editing. In *Proceedings of International Symposium on Privacy Enhancing Technologies Symposium (PETS)*. https://doi.org/10.56553/popets-2023-0080

[108] Zheng Yang and Kimmo Järvinen. 2018. The Death and Rebirth of Privacy-Preserving WiFi Fingerprint Localization with Paillier Encryption. In *Proceedings of IEEE INFOCOM*. https://doi.org/10.1109/INFOCOM.2018.8486221

[109] Lan Zhang, Cheng Bo, Jiahui Hou, Xiang-Yang Li, Yu Wang, Kebin Liu, and Yunhao Liu. 2015. Kaleido: You Can Watch It But Cannot Record It. In *Proceedings of ACM MobiCom*. https://doi.org/10.1145/2789168.2790106

[110] Lan Zhang, Taeho Jung, Cihang Liu, Xuan Ding, Xiang-Yang Li, and Yunhao Liu. 2015. POP: Privacy-Preserving Outsourced Photo Sharing and Searching for Mobile Devices. In *Proceedings of International Conference on Distributed Computing Systems (ICDCS)*. https://doi.org/10.1109/ICDCS.2015.39

[111] Ping Zhao, Hongbo Jiang, John CS Lui, Chen Wang, Fanzi Zeng, Fu Xiao, and Zhetao Li. 2018. P3-LOC: A Privacy-Preserving Paradigm-Driven Framework for Indoor Localization. *IEEE/ACM Transactions on Networking* 26, 6 (2018), 2856–2869.