

Obfuscation Based Privacy Preserving Representations are Recoverable Using Neighborhood Information - Supplementary Material

Kunal Chelani^{1*} Assia Benbihi^{2*} Fredrik Kahl¹ Torsten Sattler² Zuzana Kukelova³

¹Chalmers University of Technology

²Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague

³Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague

chelani@chalmers.se

The supplementary material is organized as follows. Sec. 1 details the point recovery from coordinate permutations [13] and how we estimate which of the coordinates is swapped to transform the recovery from coordinate permutations into a recovery from lines. Sec. 2 recalls the descriptor ambiguity in paired-point lines obfuscations [9] and how neighborhood information is used to assign descriptors to their original points. Sec. 3 reports results on the indoor 12-scenes [20] dataset as announced in Section 6. These results are consistent with the ones on the indoor 7-scenes [17] dataset. We also report the geometric and perceptual evaluation for all 3D obfuscations, including the random line obfuscation OLC [18] and the PPL+ variant of the pair-point lifting [9], that are left out of the main paper for the sake of brevity. Additionally, we also provide visual examples of the estimated neighborhood graph on two scenes from the ScanNet++ [22] dataset. Sec. 4 provides additional implementation details related to the nearest-neighbor learning and the image inversion from 2D points.

1. Coordinate permutation - Predicting swapped coordinate

As mentioned in Sec.4 of the paper, the coordinate permutation obfuscation is equivalent to obfuscating the points with multiple lines (2 in 2D, 3 in 3D) that are axes-aligned and pass through the obfuscated point $\mathcal{O}(x)$. It should be recalled that this is done for the computational feasibility of the proposed approach as explained in the main paper. Before running the proposed recovery method on these lines, we discard some of the lines so that for each point, only one of the two or three lines remains. The remaining line should follow the direction along which the point has been moved. Identifying such a line amounts to estimating which of the coordinates of the obfuscated points have been swapped.

*Equal Contribution.

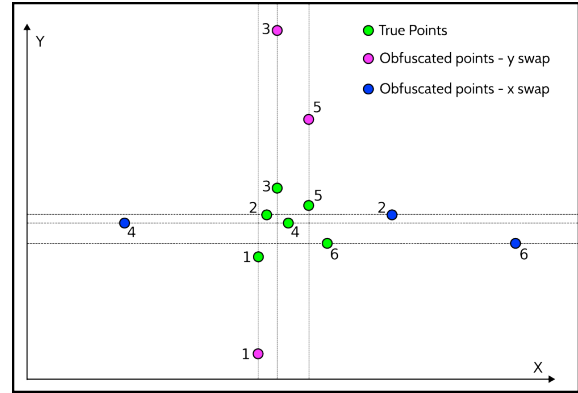


Figure 1. **Illustration of the Coordinate Swap Inversion.** The green points represent the true original points that form a neighborhood. One coordinate of each point is swapped with that of another point in the image (not shown here for brevity) to result in the blue/pink points. Note that points shifted along the y-axis (pink) form a cluster around the same x-value and similarly points shifted along the x-axis form a cluster around the same y-value. This idea is used to estimate the swapped coordinates of the members of a neighborhood.

We now describe how to identify such a line, *i.e.*, how to identify the swapped coordinate.

For each point, we predict the swapped coordinate, correspondingly the line along which the original point is estimated to lie, using neighborhood information. Our method, as illustrated in Fig. 1, is based on the observation that if one arbitrary coordinate of the points in a neighborhood is changed, then the obfuscated points (ones with swapped coordinates) remain close to each other along the remaining dimensions (the coordinates that were not swapped). In practice, given a set of obfuscated points that are known to be neighbors, we iterate through each point and compute its distances to all other points in the neighborhood along

each axis. We identify the axis which has relatively larger cumulative distances as the estimated line direction. For example, in Fig. 1, the green points show the original points in a neighborhood, and the set of blue and pink points together form the set $\mathcal{N}(\mathcal{O}(x))$ of obfuscated neighbors. Then, if we consider the blue point numbered 4, it has small distances along the y-axis to the blue points 2 and 6 but relatively larger distances along the x-axis to all points. It is therefore estimated to have been moved along the x-axis. We make this approach robust with voting, *i.e.*, we visit the neighborhood of all points and accumulate the estimated direction for each point over all visits. In the end, we select the direction with the most votes for each point.

2. Descriptor Assignment for Paired-Point-Lines

The point-paired 3D line obfuscations, PPL and PPL+ [9], transform the point cloud into a line cloud by generating lines joining random pairs of 3D points. This approach has several advantages one of which is the confusion over feature descriptors. With the line joining two points, it also holds two descriptors, each associated with one point. While the neighborhood-based recovery estimates the position of the two original points on the line, the descriptors still need to be assigned to each of the points to enable the inversion attack [15]. We provide a more formal definition of the problem and its solution.

Problem Definition: In the paired-point setting, one line holds two descriptors and the point recovery relies on two sets of neighboring lines (one for each obfuscated point). Each set of neighboring lines is used independently to estimate the position of one obfuscated point. We then want to associate each estimated point with one of the two descriptors on the line.

Solution: We first note that there is a bijection between an estimated point and a set of neighboring lines. Assigning a descriptor to an estimated point is then equivalent to assigning a descriptor to a **set** of neighboring lines. The intuition behind the proposed method is to assign each descriptor to one of the two sets of neighboring lines. To choose between the possible assignments, we assign the descriptor to the most ‘similar’ set of neighboring lines, *i.e.*, the set of lines with the most similar descriptors. We define the distance between a descriptor and a set of neighboring lines as the sum of the distances between the descriptor to be assigned and the descriptor of each neighboring line. To deal with the fact that the neighboring lines also hold two descriptors, we chose to only count the distance to the closest of the two descriptors of a given neighboring line. In practice, we compute 4 such distances between each of the two descriptors to be assigned and each of the two sets of neighboring lines. Each descriptor is assigned to one set so that the cumulative distance of the assignment is minimized.

Note that the derivation only takes as input the lines, the pair of descriptors on each line, and the neighborhood set. The position of the points, whether original or estimated, is never used.

3. Additional Results

As a reminder, the geometric evaluation measures how close the points recovered from the obfuscations are to the original points. We measure the accuracy of the recovered points as the ratio of points which Euclidean distance to the original ones is below a given threshold in cm in 3D, and in pixels in 2D. The perceptual evaluation measures how close the images inverted [15] from the points recovered from the obfuscations are to the images inverted from the original points. We report three metrics that measure the similarity between images: the Structural Similarity Index Measure (SSIM), the Peak-to-Signal Noise Ratio (PSNR), and the Learned Perceptual Image Patch Similarity [23] (LPIPS).

Geometric Evaluation of obfuscations in 2D. Tables 1 and 2 show results over the 12-scenes [20] dataset using SuperPoint [4] and SIFT [11] as the local features. The results using SIFT on 12-scenes [20] follow the same trend as the results for 7-scenes [17] shown in the main paper, with a relative difference of 1-8% in geometric accuracy. However, the geometric accuracy of the recovered points is slightly lower when using Superpoint [4] on 12-scenes [20]. We believe that this is because the keypoints are more sparsely distributed on these images: i) the 12-scenes images are larger (1296x968) than the 7-scenes [17] ones (640x480); ii) SuperPoint [4] features are typically much sparser than the SIFT [11] features, leading to a larger distance between an obfuscated point and the points in the neighborhood. Since the distance between an obfuscated points and its furthest neighbor is an upper bound on the error of the recovered point [2], a larger mean distance to the neighbors usually implies a decrease in the geometric accuracy. This suggests that one way to prevent the proposed point recovery is to use sparse keypoints but this may come at the cost of lower localization performance. Also, we observe that using a smaller neighborhood size improves the accuracy for SuperPoint [4] so sparsifying the points may not be enough since tuning the parameters of the point recovery can compensate for it. To keep the recovery parameters consistent with the rest of the paper, we show all results for 2D obfuscations using $K = 20$ as the neighborhood size.

Perceptual Evaluation of obfuscations in 2D. Figures 12 and 13 show qualitative results for images inverted in indoor scenes when using oracle-provided neighborhoods of different qualities and SIFT [11] features. It is clear that identifiable scene content is revealed even for neighborhoods of inlier ratio 0.2 in case of lifting to random lines [19]. With coordinate permutations [13], the scene remains more private and we observe that the performance bottleneck of

the point recovery lies in the preprocessing step that estimates which coordinate is swapped using the neighborhood information. Still, neighborhoods with inlier ratios of 0.5 or more are enough for the point recovery to successfully reveal the content of the scene. Figures 10 and 11 show similar results for outdoor scenes from the Cambridge [7] dataset.

Geometric Evaluation of other 3D obfuscations. In the main paper, we report results only for a subset of 3D geometric obfuscations because of the page limits: the paired-point lines PPL [9], the Ray clouds [12], the plane obfuscation [6] and the point permutation [13]. We complete these results with the evaluation of the random-line obfuscation [18] and the PPL+ variant of the paired-point lines [9] on the two indoor datasets, 7-scenes [17] and 12-scenes [20] and the outdoor dataset Cambridge [7] in Tables 3, 4, 5. The 3D models are generated with Structure-from-Motion [16] from SIFT [11] features, except for 7-scenes [17] for which additional comparisons are run with the learning-based SuperPoint [4] features.

As already observed in the main paper, the 3D line obfuscations OLC [18], PPL [9], PPL+ [9] and ray clouds [12] are the most susceptible to the point recovery, even when the neighborhood information is not reliable: more than 90% of the points can be recovered with less than 10cm errors even when only 50% of the nearest neighbor information is correct. The image inversion from points recovered with only 10% of inliers in the neighborhood still reveals the content of the original images, as can be seen in the last row of the Figures 2, 3, 4, 5. Out of the 3D line obfuscations, the most recent ray clouds appear to be the most privacy-preserving with the geometric accuracy dropping more as the inlier ratio of the neighborhoods decreases but the outline of the scene remains recognizable in the inverted images. The point recovery also works on the plane [6] and point-permutation [13] obfuscations but requires more reliable neighborhood information than for the 3D line obfuscations: the recovery is less accurate when the NN inlier ratio goes between 50% and 30%, which typically prevents meaningful image inversion.

3D Point-Paired-Line Obfuscations: PPL and PPL+ [9]. The point-paired line obfuscations, PPL and PPL+ [9], operate in 3D and transform the point cloud into a line cloud by generating lines joining random pairs of 3D points. PPL+ is an extension of PPL that discourages lines to be formed between two points that lie on the same plane for two reasons: i) such lines could give hints on the scene structure, *e.g.*, if the scene is a long corridor; ii) such lines are more vulnerable to density attacks [2] as the distribution of line distances used to derive neighbors is more characteristic around each hidden points.

In our experiments, we observe that the performance of the proposed point recovery is equivalent between PPL and

PPL+ as shown by the close geometric accuracies in Tables 3, 4, 5. These results are consistent with the original PPL paper [9] where both PPL and PPL+ are recovered with similar errors by the density-based recovery [2]. One advantage of PPL over PPL+, though, is its faster runtime: PPL+ keeps drawing point-paired lines as long as the plane condition is not satisfied or until a certain number of iterations is reached. When PPL can terminate in a matter of minutes on a small indoor point cloud typical of 12 scenes [20], PPL+ can take several hours.

Perceptual Evaluation of other 3D obfuscations. In the main paper, we reported only SSIM for the sake of clarity since the three metrics exhibit the same trend over all obfuscations and inlier ratios. For the sake of completeness, we additionally report the SSIM and PSNR values for all 3D obfuscations on 7-scenes [17] (Table 6), 12-scenes [20] (Table 7), and Cambridge [7] (Table 8). To keep the table readable, we report values only for PPL [9] as the PPL+ [9] perceptual metrics are either equal or within 0.01 difference, which is negligible.

Similarly to the geometric evaluation, the recovery from the line obfuscations OLC [18], PPL [9] and ray clouds [12] is stable across the inlier ratio of the neighborhood information whereas the recovery from the plane [6] and the point permutation [13] is more sensitive to incorrect neighbors between 50% and 30% inlier ratios.

Comparison to other 3D line recoveries. We compare the proposed point recovery to the existing density-based recoveries in [2] and [9] that operate on 3D lines only (Table 9). These methods estimate the neighborhood of a given 3D line based on the density of all lines in the cloud and the original point is approximated with the position of highest density along the line. We observe that our method largely outperforms those baselines even with as little as 20% inlier ratio in the neighborhood information necessary for our recovery.

However, we note that the results for [9] computed with the author’s public release seem subpar to the results reported in the paper so this comparison should be taken as an indicative result only. We believe that this discrepancy in the results is not due to a technical issue in the method or the code of [9] but rather the difference in input data: the point clouds we generated and the points clouds of [9] are most likely different because of variations in the Structure-from-Motion [16], *e.g.*, because of differences in the parameters or the randomness of the robust geometric estimation. To reduce the potential discrepancy in the input data and for this experiment only, we use the point clouds used in [2] to run this evaluation instead of the point clouds we generated for the rest of the paper. However, discrepancies between the input data used in [2] and [9] remain and this is why these results should be taken as indicative results only.

Influence of the features on the point recovery in 3D. We

In.	SuperPoint [4]						SIFT [11]					
	CP [13]			Lines [19]			CP [13]			Lines [19]		
	5px	10px	25px	5px	10px	25px	5px	10px	25px	5px	10px	25px
1.0	12.9	24.43	52.8	13.5	26.4	56.6	42.7	67.2	89	41.3	66.8	89.6
0.75	14.5	27.4	56.5	15.8	30.7	62.42	43.8	69.1	89.7	42.9	69.6	91.5
0.50	14.4	26.4	52.4	18.5	35.6	69.1	40	64	82.1	43.7	71.7	93.3
0.30	8.68	15.98	30.72	19.7	38.0	71.6	20.1	33.1	43.9	41.5	69.8	91.9
0.20	4.93	9.1	17.7	17.5	33.8	63.0	9.0	14.9	21.8	34.2	58.6	79.0
0.10	2.35	4.39	9.06	9.09	17.4	32.7	3.2	5.5	9.7	15	25.5	37.7

Table 1. **Geometric accuracy of the point recovery from 2D obfuscations** on the 12-scenes [20] dataset using two different features : SuperPoint [4] and SIFT [11]. The geometric accuracy, *i.e.*, the fraction of recovered points with an error lower than a given threshold, is lower in general as compared to results over 7Scenes [17] because of the larger image sizes in the 12-scenes [20] dataset - 1296x968 as compared to 640x480. Further, SuperPoint [4] features are typically much sparser than SIFT features, increasing the average distance to neighbors. The average number of SuperPoint [4] features per image in our experiment was around 312 as compared to 1412 for SIFT [11]. This suggests that one way to prevent the proposed point recovery is to use sparse keypoints but this may come at the cost of a lower localization performance.

In.	SuperPoint [4]						SIFT [11]					
	CP [13]			Lines [19]			CP [13]			Lines [19]		
	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑
Baseline	0.55	0.48	15.5	0.55	0.48	15.5	0.60	0.55	14.6	0.60	0.55	14.6
1.0	0.42	0.6	13.5	0.42	0.59	13.8	0.52	0.63	14	0.51	0.64	14.1
0.75	0.41	0.60	13.4	0.42	0.59	13.9	0.51	0.64	13.9	0.51	0.64	14.1
0.50	0.39	0.61	12.9	0.43	0.58	14.1	0.49	0.66	13.3	0.52	0.63	14.3
0.30	0.37	0.63	12.2	0.43	0.58	14.0	0.43	0.70	12.0	0.52	0.64	14.2
0.20	0.37	0.64	11.9	0.41	0.6	13.7	0.41	0.72	11.5	0.49	0.65	13.6
0.10	0.36	0.65	11.6	0.35	0.63	12.8	0.41	0.73	11.2	0.41	0.70	11.9

Table 2. **Perceptual accuracy of the point recovery from 2D obfuscations** on the 12-scenes [20] dataset using SuperPoint [4] and SIFT [11]. **Baseline** refers to the similarity score between the real image and the image inverted from the original points. The results follow the same trends as that for 7-scenes [17] shown in the main paper.

assess whether the performance of the point recovery depends on the type of features extracted from the images and used for the Structure-from-Motion [16] that generates the 3D point cloud. We compare the hand-crafted SIFT [11] and the deep-learning-based SuperPoint [4] and report the geometric accuracy on 7-scenes [17] in Tab. 3. The performance of the point recovery is consistent between the 3D models generated from SuperPoint [4] and SIFT [11] with variations in geometric accuracy in the order of a few percent. This shows that the method is insensitive to the features used to generate the 3D model, which is not that surprising given that the optimization in the point recovery relies on the geometry only.

Qualitative Results. Further examples of images inverted from the points recovered from various obfuscations are shown in Figures 2, 3, 4, 5 on 7scenes [17], and in Figures 6, 7, 8, 9 on Cambridge [7].

Detected content. In addition to the previous perceptual evaluation, we measure the recovered information at the finer level of objects and adopt the SegLoc’s evaluation [14]. An off-the-shelf object detector, YoloV7 [21], runs on both the images inverted from the original points and the recovered points. The discrepancy between the two sets of detections is a relevant proxy to measure how much content is recovered. We report the standard detection metric in Table 11 where the detections on the real images are used as ground-truth and the detections on the images inverted from the original points clouds are the baseline. For the sake of clarity, we only report here the recall of the detection for it indicates the amount of objects discovered by the attack, which is more relevant than the precision at which the object is discovered. These values are indicative only as when we appraise the inverted images visually, it often occurs that the inverted images is decipherable by the

SIFT [11]		OLC [18]		PPL [9]		PPL+ [9]		Rays [12]		Plane [6]		CP [13]	
	In.	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm
	1.0	96.1	98.6	94.6	97.3	94.8	97.4	94.6	97.9	93.4	97.5	88.2	94.5
	0.75	96.0	98.2	94.7	97.1	94.9	97.3	93.3	96.8	93.0	97.0	89.1	95.8
	0.50	96.2	98.2	95.0	97.2	95.1	97.3	91.9	95.7	82.8	88.7	67.7	75.0
	0.30	96.4	98.2	94.8	97.1	94.9	97.2	86.2	90.5	42.1	60.4	40.9	46.2
	0.20	96.3	98.2	94.0	96.8	94.1	96.9	78.7	83.6	20.9	39.6	31.1	35.1
	0.10	92.5	96.1	78.2	84.5	78.3	84.7	49.9	57.1	7.5	20.7	22.8	26.2
SuperPoint [4]		OLC [18]		PPL [9]		Rays [12]		Plane [6]		CP [13]			
	In.	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm
	1.0	98.3	99.7	96.9	99.0	94.7	98.2	95.6	99.0	89.8	96.1		
	0.75	98.2	99.6	97.2	99.0	93.3	97.2	95.1	98.8	90.4	97.6		
	0.50	98.5	99.6	97.3	99.1	91.9	96.2	82.3	90.5	66.5	74.8		
	0.30	98.6	99.6	96.6	98.8	85.8	91.1	40.8	63.4	39.8	45.8		
	0.20	98.5	99.6	94.8	97.8	77.7	83.8	21.8	44.1	30.4	35.3		
	0.10	93.5	97.0	72.7	80.5	47.3	55.9	9.6	28.3	21.7	26.0		

Table 3. **Geometric accuracies \uparrow of the 3D point recovery on the indoor 7-scenes [17] with SIFT [11] and SuperPoint [4].** The point clouds are generated with Structure-from-Motion [16]. The performance of the point recovery is consistent between the 3D models generated from SuperPoint and SIFT [11] with variations in geometric accuracy in the order of a few percent, up to 8% with the worst inlier ratio of 0.1. This shows that the method is insensitive to the features used to generate the 3D model, which is not that surprising given that the optimization in the point recovery relies on the geometry only. The line obfuscations OLC [18], PPL [9], PPL+ [9] and Ray clouds [12] are the most susceptible to the recovery, even when the neighborhood information is not reliable. The point recovery also works on the plane [6] and point-permutation [13] obfuscations but requires more reliable neighborhood information than for the previous obfuscation.

	OLC [18]		PPL [9]		PPL+ [9]		Rays [12]		Plane [6]		CP [13]	
In.	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm	10cm	25cm
1.0	99.2	99.8	98.8	99.6	98.4	99.1	97.7	99.2	99.0	99.7	92.3	96.2
0.75	99.3	99.8	98.8	99.6	98.9	99.6	95.0	97.7	97.4	98.2	94.2	98.9
0.50	99.3	99.8	98.6	99.6	98.8	99.6	93.3	97.3	79.3	84.2	72.7	79.7
0.30	99.4	99.8	97.8	99.5	98.2	99.5	91.5	96.5	38.5	54.0	39.6	44.2
0.20	99.3	99.8	96.0	98.7	96.8	99.0	88.4	94.0	20.0	36.1	27.6	31.0
0.10	98.6	99.7	85.6	90.8	87.5	92.1	71.1	78.4	8.0	20.2	18.5	21.8

Table 4. **Geometric accuracies \uparrow of the 3D point recovery on the indoor 12-scenes [20] with SIFT [11].** The conclusions are consistent with the results on the other indoor dataset 7-scenes [17] reported in Table 3: the line obfuscations OLC [18], PPL [9], PPL+ [9] and Ray clouds [12] are the most susceptible to the recovery, even when the neighborhood information is not reliable (*e.g.* 10%), whereas the plane [6] and point-permutation [13] are not recovered reliably as soon as the inlier ratio in the neighborhood information drops.

human eye but the detection fails to identify the objects because of the domain shift and the noise of the image. Hence, the detection performance tends to over-estimate the privacy of the evaluated representations.

4. Implementation Details

Geometric Recovery and Runtime. The point recovery runs within a reasonable amount of time: the minimiza-

tion is implemented using the open-source Ceres [1] optimization library and runs in parallel on a single CPU. The runtime is a function of the number of points in the point cloud or the image, the inlier ratio, the neighborhood size, and the maximum number of RANSAC [5] iterations: the more points and the larger the neighborhood, the more time the computation takes. In parallel, the higher the inlier ratio, the lower the runtime as the optimal number of

	OLC [18]		PPL [9]		PPL+ [9]		Rays [12]		Plane [6]		Perm. [13]	
In.	25cm	50cm	25cm	50cm	25cm	50cm	25cm	50cm	25cm	50cm	25cm	50cm
1.0	74.4	87.6	69.2	83.2	69.5	83.3	72.1	83.6	65.2	81.1	65.3	81.0
0.75	71.6	84.7	66.9	80.4	67.7	80.8	72.9	83.1	56.2	67.7	66.3	82.0
0.50	71.6	83.4	67.2	79.3	67.7	79.6	74.4	84.1	33.2	38.5	61.4	72.5
0.30	72.5	83.2	68.2	78.8	68.5	79.0	75.5	84.8	15.0	17.1	35.4	40.6
0.20	73.5	83.1	69.0	78.4	69.2	78.6	75.0	84.2	8.1	9.4	24.1	27.2
0.10	72.7	80.2	69.1	76.2	69.3	76.4	63.8	72.7	2.9	3.8	16.5	18.2

Table 5. **Geometric accuracies \uparrow of the 3D point recovery on the outdoor Cambridge [7] dataset with SIFT [11].** The same trend is observed outdoors as it is indoors, *i.e.*, the line obfuscations OLC [18], PPL [9], PPL+ [9] and Ray clouds [12] are the most susceptible to the recovery, even when the neighborhood information is not reliable (*e.g.* 10%), whereas the plane [6] and point-permutation [13] are not recovered reliably when the inlier ratio drops too low. Although the geometric accuracy values are lower than for indoor and measured at higher error thresholds, the images inverted from the recovered points remain meaningful as shown in Figures 6, 7, 8, 9.

	LPIPS \downarrow Baseline: 0.52					SSIM \uparrow Baseline: 0.58					PSNR \uparrow Baseline: 16.01				
In.	OLC	PPL	Ray	Plane	CP	OLC	PPL	Ray	Plane	CP	OLC	PPL	Ray	Plane	CP
1.0	0.53	0.53	0.53	0.55	0.55	0.58	0.57	0.57	0.55	0.56	15.9	15.8	15.8	15.5	15.6
0.75	0.53	0.54	0.54	0.56	0.55	0.57	0.57	0.57	0.54	0.55	15.9	15.7	15.8	15.3	15.5
0.50	0.53	0.55	0.54	0.60	0.59	0.57	0.56	0.56	0.49	0.51	15.8	15.6	15.7	14.4	14.7
0.30	0.54	0.55	0.55	0.64	0.63	0.57	0.55	0.56	0.44	0.45	15.8	15.4	15.5	13.0	13.1
0.20	0.54	0.56	0.56	0.66	0.65	0.57	0.54	0.54	0.43	0.43	15.8	15.2	15.3	12.1	12.6
0.10	0.54	0.59	0.60	0.68	0.66	0.56	0.51	0.50	0.42	0.41	15.7	14.4	14.3	11.5	12.2

Table 6. **Perceptual metrics of the 3D point recovery on the indoor 7-scenes [17] with SIFT [11]:** the metrics assess how close the image inverted [15] from the points recovered from the obfuscations is to the image inverted [15] from the original points. As for the geometric evaluation, the recovery from the line obfuscations OLC [18], PPL [9] and Ray cloud [12] is stable across the inlier ratio of the neighborhood information whereas the recovery from the plane [6] and the point permutation [13] is more sensitive to incorrect neighbors.

RANSAC [5] iterations is inversely proportional to the inlier ratio. For example, the biggest point cloud in the experiments has 700K points (12-scenes-office1-gates-381 [20]). In a setup with 100 neighbors and an upper bound on the number of RANSAC [5] iterations set to 10K, the runtime varies between 1 minute 30 s when there are no outliers in the neighborhood up to 4 minutes for inlier ratios between 75% and 20%, on a single AMD EPYC CPU with 64 cores. Table 10 gives more runtime examples as a function of the point cloud size and inlier ratios.

Point Initialization. We observe that the point recovery is insensitive to the point initialization and use the following heuristics in the paper.

For 3D lines [9, 12, 18] and 3D lines made from 3D permutation [13], the 3D points to recover are initialized as the projection of a 3D point "anchor" onto the lines. The 3D anchor is defined as follows: the 3D lines are projected onto a plane. We sample a set of intersections between the resulting 2D lines and compute their 2D centroid, which we use as the anchor. In the paper, we use the plane $z = 0$

so the centroid has the form $(x, y, 0)$ and randomly sample 10K intersection points. Note that the choice for the plane $z = 0$ does not necessarily correspond to the ground-plane as the coordinate frames of the scenes are chosen arbitrarily by the authors of the datasets.

The initialization in 2D follows the same steps except that the 2D lines already intersect so there is no need to project them onto a plane.

For planes, we also project a 3D "anchor" point onto each plane but the anchor point is built differently: it is defined as the 3D point which coordinates are the average of the planes' offsets associated with that axis, *i.e.*, the x coordinate is the average of offsets c of all planes of the form $x = c$.

NN Learning. The recovery of the points hidden by obfuscated representations assumes that the original points' neighborhood information is available, *i.e.*, one knows which obfuscations hide points that are close to each other. The main experiments are run using an oracle that produces neighborhoods with various levels of inlier ratios to allow

In.	LPIPS↓ Baseline: 0.52					SSIM↑ Baseline: 0.58					PSNR↑ Baseline: 16.01				
	OLC	PPL	Ray	Plane	CP	OLC	PPL	Ray	Plane	CP	OLC	PPL	Ray	Plane	CP
1.0	0.56	0.57	0.57	0.58	0.58	0.49	0.49	0.48	0.48	0.47	14.6	14.5	14.3	14.3	14.3
0.75	0.57	0.57	0.58	0.60	0.58	0.49	0.48	0.48	0.44	0.47	14.6	14.5	14.3	13.9	14.2
0.50	0.57	0.58	0.58	0.64	0.61	0.49	0.48	0.47	0.40	0.43	14.5	14.4	14.3	12.9	13.3
0.30	0.57	0.58	0.58	0.66	0.65	0.49	0.47	0.47	0.36	0.38	14.5	14.3	14.2	11.8	12.1
0.20	0.57	0.59	0.58	0.67	0.66	0.49	0.47	0.47	0.35	0.36	14.5	14.2	14.2	11.3	11.6
0.10	0.57	0.60	0.60	0.69	0.67	0.49	0.45	0.45	0.34	0.34	14.5	13.8	13.8	10.8	11.2

Table 7. **Perceptual metrics of the 3D point recovery on the indoor 12-scenes [20] with SIFT [11]**: the metrics assess how close the image inverted [15] from the points recovered from the obfuscations is to the image inverted [15] from the original points. As for the geometric evaluation, the recovery from the line obfuscations OLC [18], PPL [9] and Ray cloud [12] is stable across the inlier ratio of the neighborhood information whereas the recovery from the plane [6] and the point permutation [13] is more sensitive to incorrect neighbors.

In.	LPIPS↓ Baseline: 0.52					SSIM↑ Baseline: 0.58					PSNR↑ Baseline: 16.01				
	OLC	PPL	Ray	Plane	CP	OLC	PPL	Ray	Plane	CP	OLC	PPL	Ray	Plane	CP
1.0	0.64	0.64	0.63	0.64	0.64	0.37	0.36	0.37	0.36	0.36	12.8	12.7	12.8	12.7	12.7
0.75	0.64	0.64	0.63	0.66	0.64	0.36	0.36	0.37	0.34	0.36	12.7	12.6	12.8	12.2	12.7
0.50	0.64	0.64	0.63	0.67	0.66	0.36	0.36	0.37	0.32	0.34	12.6	12.6	12.8	11.6	12.2
0.30	0.64	0.64	0.63	0.69	0.69	0.36	0.36	0.37	0.31	0.29	12.6	12.5	12.8	10.9	11.2
0.20	0.64	0.65	0.64	0.70	0.70	0.36	0.36	0.37	0.31	0.27	12.5	12.5	12.7	10.5	10.8
0.10	0.65	0.65	0.65	0.71	0.70	0.34	0.35	0.35	0.30	0.26	12.3	12.4	12.4	10.3	10.6

Table 8. **Perceptual metrics of the 3D point recovery on the outdoor Cambridge [7] with SIFT [11]**: the metrics assess how close the image inverted [15] from the points recovered from the obfuscations is to the image inverted [15] from the original points. Similarly to the geometric evaluation, the recovery from the line obfuscations OLC [18], PPL [9], and Ray clouds [12] is stable across the inlier ratio of the neighborhood information whereas the recovery from the plane [6] and the point permutation [13] is more sensitive to incorrect neighbors.

Recovery	OLC 3D lines	PPL 3D lines
	5 / 10 / 25 cm	5 / 10 / 25 cm
OLC Rec. [2]	67.5 / 75.8 / 84.0	—
PPL Rec. [9]	—	34.85 / 48.71 / 63.16
Ours 50% In.	94.6 / 99.3 / 99.9	89.7 / 98.2 / 99.5
Ours 20% In.	91.7 / 99.1 / 99.8	82.1 / 93.9 / 97.4

Table 9. **Comparison against 3D line recovery baselines**. Geometric accuracy ↑ of the recovery from 3D obfuscations against baseline methods [2, 9] on 12-scenes. The recoveries are run on the same 12scenes [20] point clouds as in [2], which differ from the point clouds used in the rest of the paper that we generated ourselves with COLMAP [16] or from the points clouds from [9].

Num. Pts	In 1.0	In 0.50	In 0.30	In 0.20	In 0.10
700K	0:41	0:51	1:30	4:00	18:00
300K	0:19	0:24	0:45	1:20	9:00
100K	0:08	0:10	0:19	0:46	3:40

Table 10. **Indicative runtime** as a function of the number of 3D points (Num.Pts) and inlier ratios (In.) for the recovery from the PPL [9] obfuscation with 50 neighbors. The 3D points cloud is generated from SfM [16] on SIFT [11] features. X:Y indicates that the runtime takes X minutes and Y seconds. The theoretical number of RANSAC [5] in the optimization is inversely proportional to the inlier ratio, hence the longer runtimes as the inlier ratio decreases. Still, the runtime remains small enough that the point recovery is practical for an attacker. The recovery runs on a single AMD EPYC CPU with 64 cores.

for the evaluation of the robustness of the recovery against inaccurate neighborhood information. In parallel, we show that the descriptors preserved by the geometric obfuscation hold enough information to infer the neighborhood necessary for the recovery. To do so, we train a transformer-based network to learn a similarity score between all pairs of descriptors that is inversely proportional to the distance be-

tween the original points. A simple nearest-neighbor search using the learned similarity lets us infer nearest-neighbor points.

The network is made of 6 self-attention blocks with 4 heads. Prior to being fed to the attention blocks, the

Object	3D				2D		
	Baseline	PPL	Plane	Perm.	Baseline	Line	Perm.
TV	19.5	11.4 / 15.3	5.6 / 13.5	7.1 / 13.6	16.0	7.0 / 5.8	2.4 / 5.0
Backpack	21.1	11.7 / 17.5	3.6 / 5.1	8.8 / 8.0	14.6	1.5 / 0.7	0 / 1.4
Plant	23.0	10.5 / 25.4	5.7 / 25.4	5.7 / 22.9	34.4	17.2 / 13.1	9.8 / 16.4

Table 11. **Private content detected** on the images inverted from the recovered point. The original points are derived from SIFT [11]. The detection [21] on the original images serves as ground-truth and the baseline indicates the performance of the detection on the images inverted from the original points. We report the detection recall \uparrow on points recovered from neighborhood information at inlier ratios (0.50 / 1.0). Even though the recall of the images inverted from obfuscations is lower than the baseline, we observe that this evaluation under-estimate the amount of private content that is revealed. This is because the off-the-shelf detector is typically subpar on the inverted images: it fails to detect objects that the human eye can still perceive, which is usually because of the distribution shift in the image pixels or because of the noise in the image.

input descriptor is first projected onto a 256-dimension space with an MLP. The SIFT-variant of the network is trained on SIFT [11] features extracted from 97K images sampled from all the scenes of the ScanNet dataset [3]. The SuperPoint-variant of the network is trained on Superpoint [11] features extracted from 309K images sampled from 184 scenes of the ScanNet dataset [3]. The network is trained with a batch size of 16 for 10 epochs, with the Adam [8] optimizer with an initial learning rate of $5 \cdot 10^{-4}$ with a learning rate decay of 10% starting the 3rd epoch and stopping once the learning rate reaches 10^{-5} .

Image inversion from Points. We used different inversion networks on the 2D and 3D structures to generate the images from the recovered points. **In 3D**, we use the off-the-shelf inversion network provided by Pittaluga *et al.* [15]. **In 2D**, we train a new model with the CoarseNet and RefineNet models of [15] in conjunction. The input to the network is a set of keypoints with associated descriptors only. A loss function that fuses the L1 pixel loss and the LPIPS [23] perceptual loss is used, with 0.2 and 0.8 as their corresponding weights. We train two different variants for indoor and outdoor scenes. The indoor model is trained on 200 scenes from the ScanNet [3] dataset and the outdoor variant is trained on 150 scenes of Megadepth [10]. Note that we do not use any of these two datasets for any evaluation.

Visualizing Estimated Neighbors. As an additional way to evaluate the quality of the estimated neighborhood, we draw the neighborhood graph on top of the images in Figures 14 and 15. SuperPoint [4] keypoints form the nodes of the graph and the graph has an edge between each point and its top-5 nearest neighbors estimated by our neighborhood estimation network (Sec. 5 of the main paper). We use two scenes taken from the ScanNet++ [22] dataset showing a bedroom and an office. The accuracy of our proposed point position estimation depends on the distance of the nearest neighbors used – the error in estimation increases if points

that are far apart are considered as neighbors. We therefore color the edges green if the distance between the corresponding nodes is less than a threshold and red otherwise. We use $\epsilon = 0.1 * \min(h, w)$ as the threshold where h, w are the height and width of the image. It is worth noting that in regions of images with more texture, such as texts, paintings, and other distinct objects, the keypoint density is high and the quality of estimated neighbors is also high. These are regions typically containing private user content. In texture-less parts of the scene, such as floors, walls and ceilings, the keypoints are sparse and their local regions are visually less distinct, making neighborhood estimation difficult, as illustrated by several red edges. However, often such regions do not contain information that is private to the user. Measures such as SSIM and PSNR treat all parts of the image equally, whereas from a privacy point of view, recovering certain parts of the image with good detail is enough to deem the method as not privacy-preserving. More nuanced methods to measure the privacy aspect of inverted images are therefore needed.

References

- [1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 2023. 5
- [2] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15668–15678, 2021. 2, 3, 7
- [3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017. 8
- [4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR workshops*, pages 224–236, 2018. 2, 3, 4, 5, 8, 21, 22
- [5] Martin A Fischler and Robert C Bolles. Random sample

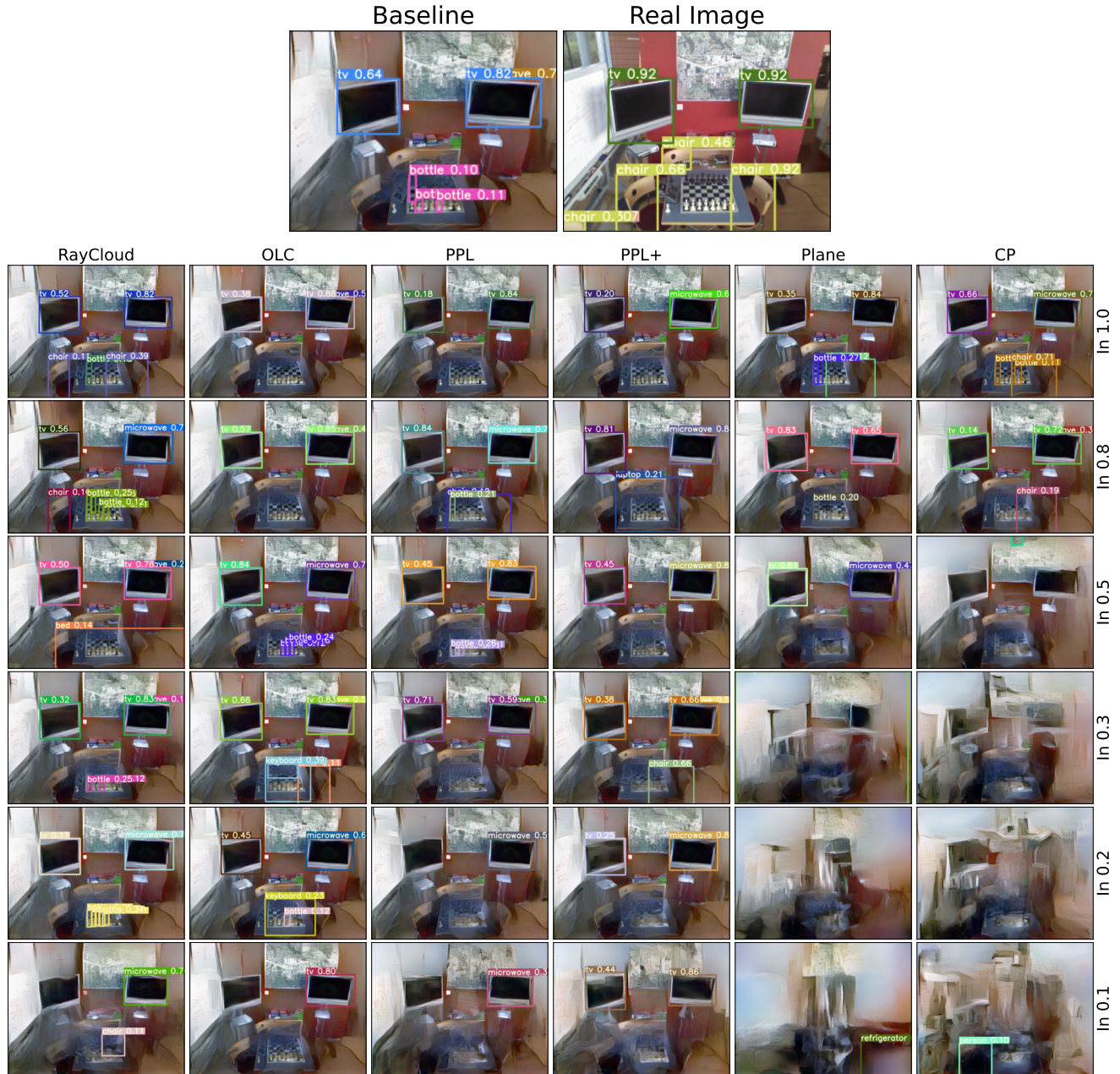


Figure 2. **Additional Qualitative Results - 7-scenes [17]-Chess.** Images inverted [15] from the original points (‘Baseline’) and the points recovered from the 3D obfuscations from neighborhood information with various levels of inlier ratios (In.). Line obfuscations (OLC) [18, 19], Point-Pair-Lines PPL and PPL+ [9], and ray clouds [12] are the most vulnerable to neighborhood-based attacks while Planes [6] and Permutations [13] are more privacy preserving. The 3D points cloud is generated from SfM [16] on SIFT [11] features.

consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5, 6, 7

- [6] Marcel Geppert, Viktor Larsson, Johannes L Schönberger, and Marc Pollefeys. Privacy preserving partial localization. In *CVPR*, pages 17337–17347, 2022. 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16

- [7] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 3, 4, 6, 7, 13, 14, 15, 16, 18

- [8] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on*



Figure 3. **Additional Qualitative Results - 7-scenes [17]-Redkitchen.** Images inverted [15] from the original points (‘Baseline’) and the points recovered from the 3D obfuscations from neighborhood information with various levels of inlier ratios (In.). Line obfuscations (OLC) [18, 19], Point-Pair-Lines PPL and PPL+ [9] and ray clouds [12] are the most vulnerable to neighborhood-based attacks while Planes [6] and Permutations [13] are more privacy preserving. The 3D points cloud is generated from SfM [16] on SIFT [11] features.

Learning Representations (ICLR), San Diego, CA, USA, 2015. 8

- [9] Chunghwan Lee, Jaihoon Kim, Chanhyuk Yun, and Je Hyeong Hong. Paired-point lifting for enhanced privacy-preserving visual localization. In *CVPR*, pages 17266–17275, 2023. 1, 2, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16

- [10] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos, 2018. 8

- [11] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

- [12] Heejoon Moon, Chunghwan Lee, and Je Hyeong Hong. Efficient privacy-preserving visual localization using 3d ray



Figure 4. **Additional Qualitative Results - 7-scenes [17]-Office.** Images inverted [15] from the original points (‘Baseline’) and the points recovered from the 3D obfuscations from neighborhood information with various levels of inlier ratios (In.). Line obfuscations (OLC) [18, 19], Point-Pair-Lines PPL and PPL+ [9], and ray clouds [12] are the most vulnerable to neighborhood-based attacks while Planes [6] and Permutations [13] are more privacy preserving. The 3D points cloud is generated from SfM [16] on SIFT [11] features.

clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9773–9783, 2024. 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16

- [13] Linfei Pan, Johannes L Schönberger, Viktor Larsson, and Marc Pollefeys. Privacy preserving localization via coordinate permutations. In *ICCV*, pages 18174–18183, 2023. 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16

- [14] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. Segloc: Learning segmentation-based representations for privacy-preserving visual localization. In *CVPR*, pages 15380–15391, 2023. 4

- [15] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *CVPR*, pages 145–154,

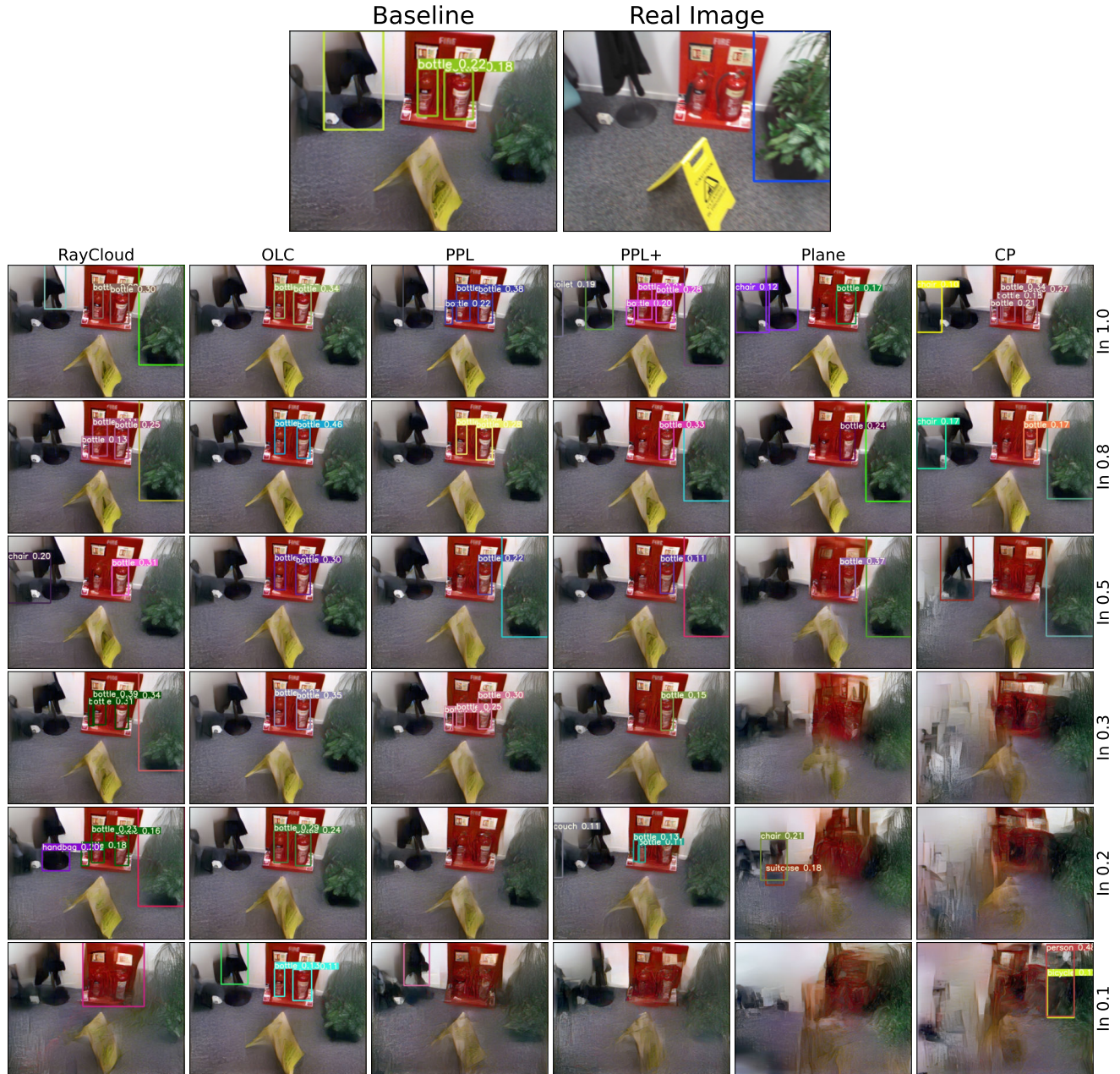


Figure 5. **Additional Qualitative Results - 7-scenes [17] dataset, scene Fire.** Images inverted [15] from the original points (‘Baseline’) and the points recovered from the 3D obfuscations from neighborhood information with various levels of inlier ratios (In.). Line obfuscations (OLC) [18, 19], Point-Pair-Lines PPL and PPL+ [9], and ray clouds [12] are the most vulnerable to neighborhood-based attacks while Planes [6] and Permutations [13] are more privacy preserving. The 3D points cloud is generated from SfM [16] on SIFT [11] features.

2019. 2, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

- [16] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16
- [17] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene co-

ordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, pages 2930–2937, 2013. 1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 17

- [18] Pablo Speciale, Johannes L Schonberger, Sing Bing Kang, Sudepta N Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,



Figure 6. **Additional Qualitative Results - Cambridge [7] dataset, scene *Shop Facade*.** Images inverted [15] from the original points (‘Baseline’) and the points recovered from the 3D obfuscations from neighborhood information with various levels of inlier ratios (In.). Line obfuscations (OLC) [18, 19], Point-Pair-Lines PPL and PPL+ [9] and ray clouds [12] are the most vulnerable to neighborhood-based attacks while Planes [6] and Permutations [13] are more privacy preserving. The 3D points cloud is generated from SfM [16] on SIFT [11] features.

- pages 5493–5503, 2019. 1, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16
- [19] Pablo Speciale, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image queries for camera localization. In *ICCV*, pages 1486–1496, 2019. 2, 4, 9, 10, 11, 12, 13, 14, 15, 16
- [20] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*, pages 323–332. IEEE, 2016. 1, 2, 3, 4, 5, 6, 7, 19, 20
- [21] CY Wang, A Bochkovskiy, and HYM Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arxiv* 2022. *arXiv preprint arXiv:2207.02696*, 2022. 4, 8
- [22] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 1, 8, 21, 22
- [23] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2, 8



Figure 7. **Additional Qualitative Results - Cambridge [7] dataset, scene *King's College***. Images inverted [15] from the original points ('Baseline') and the points recovered from the 3D obfuscations from neighborhood information with various levels of inlier ratios (In.). Line obfuscations (OLC) [18, 19], Point-Pair-Lines PPL and PPL+ [9], and ray clouds [12] are the most vulnerable to neighborhood-based attacks while Planes [6] and Permutations [13] are more privacy preserving. The 3D points cloud is generated from SfM [16] on SIFT [11] features.



Figure 8. **Additional Qualitative Results - Cambridge [7] dataset, scene *Old Hospital*.** Images inverted [15] from the original points (‘Baseline’) and the points recovered from the 3D obfuscations from neighborhood information with various levels of inlier ratios (In.). Line obfuscations (OLC) [18, 19], Point-Pair-Lines PPL and PPL+ [9], and ray clouds [12] are the most vulnerable to neighborhood-based attacks while Planes [6] and Permutations [13] are more privacy preserving. The 3D points cloud is generated from SfM [16] on SIFT [11] features.

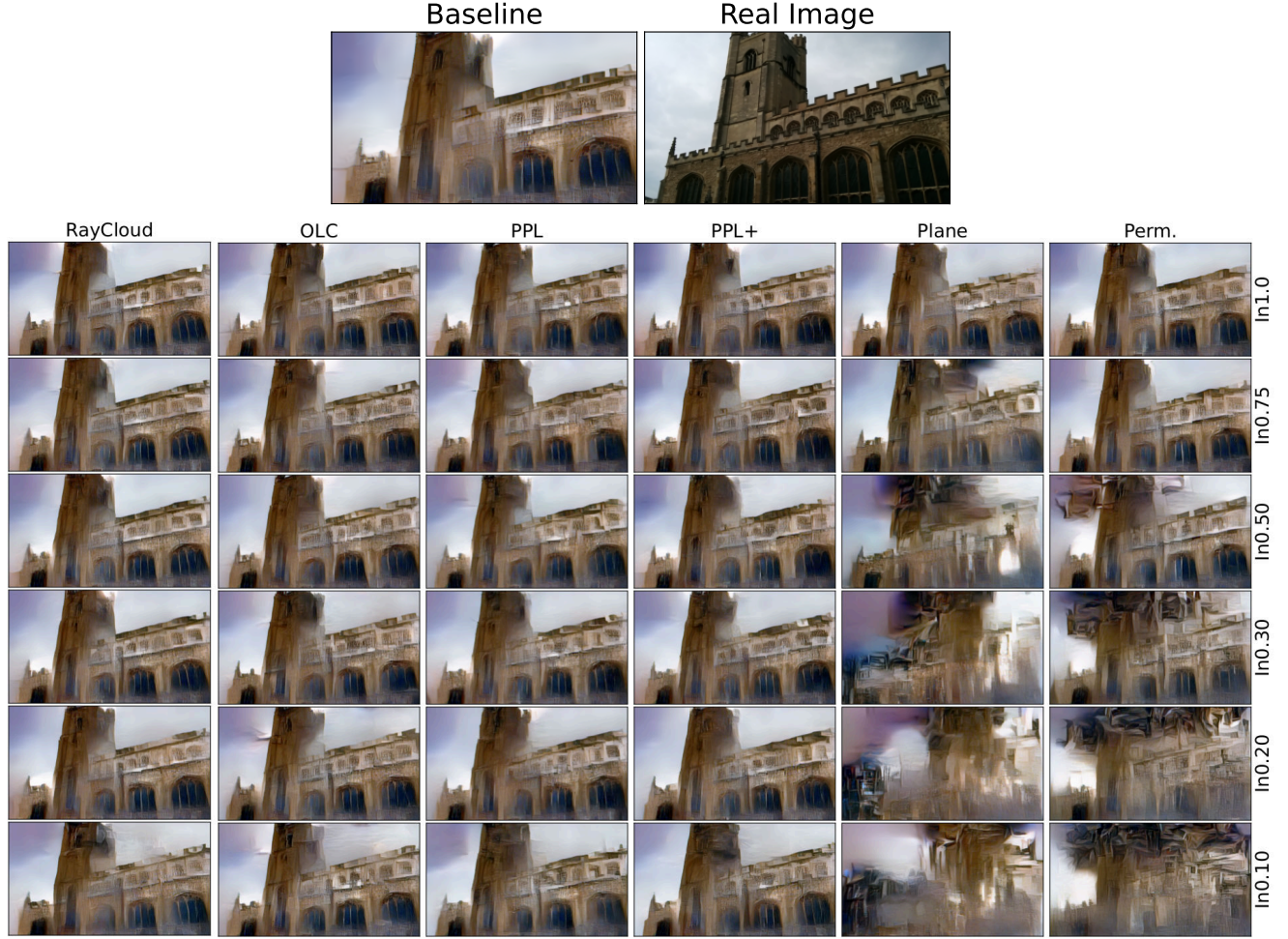


Figure 9. **Additional Qualitative Results - Cambridge [7] dataset, scene *St. Mary's Church*.** Images inverted [15] from the original points ('Baseline') and the points recovered from the 3D obfuscations from neighborhood information with various levels of inlier ratios (In.). Line obfuscations (OLC) [18, 19], Point-Pair-Lines PPL and PPL+ [9], and ray clouds [12] are the most vulnerable to neighborhood-based attacks while Planes [6] and Permutations [13] are more privacy preserving. The 3D points cloud is generated from SfM [16] on SIFT [11] features. The 3D points cloud is generated from SfM [16] on SIFT [11] features.

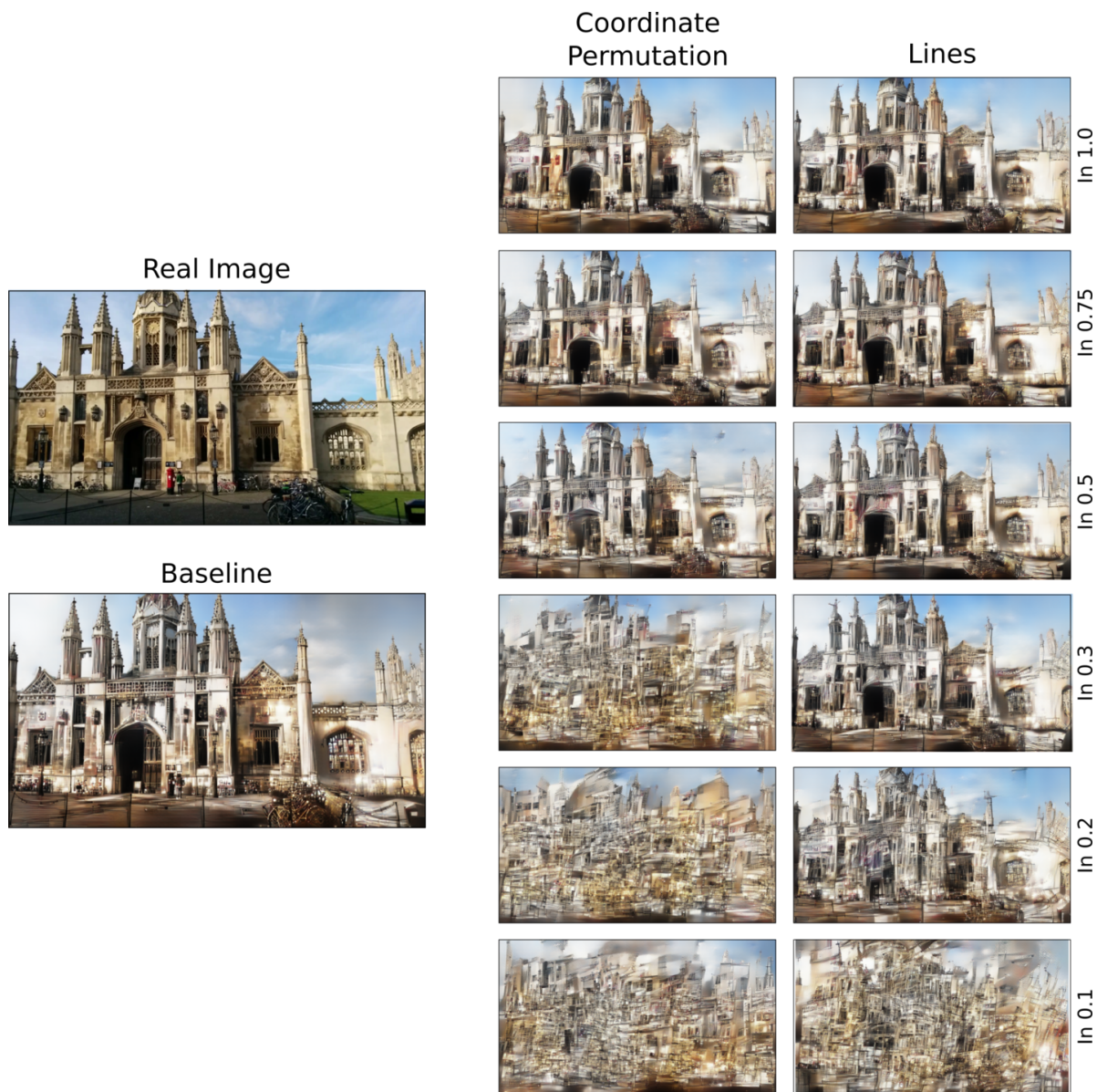


Figure 10. **Additional Qualitative Results - Cambridge [17] dataset, scene *King's College*.** Images inverted from the original 2D SIFT [11] keypoints and the points recovered from the 2D obfuscations using neighborhood information with various levels of inlier ratios (In.).

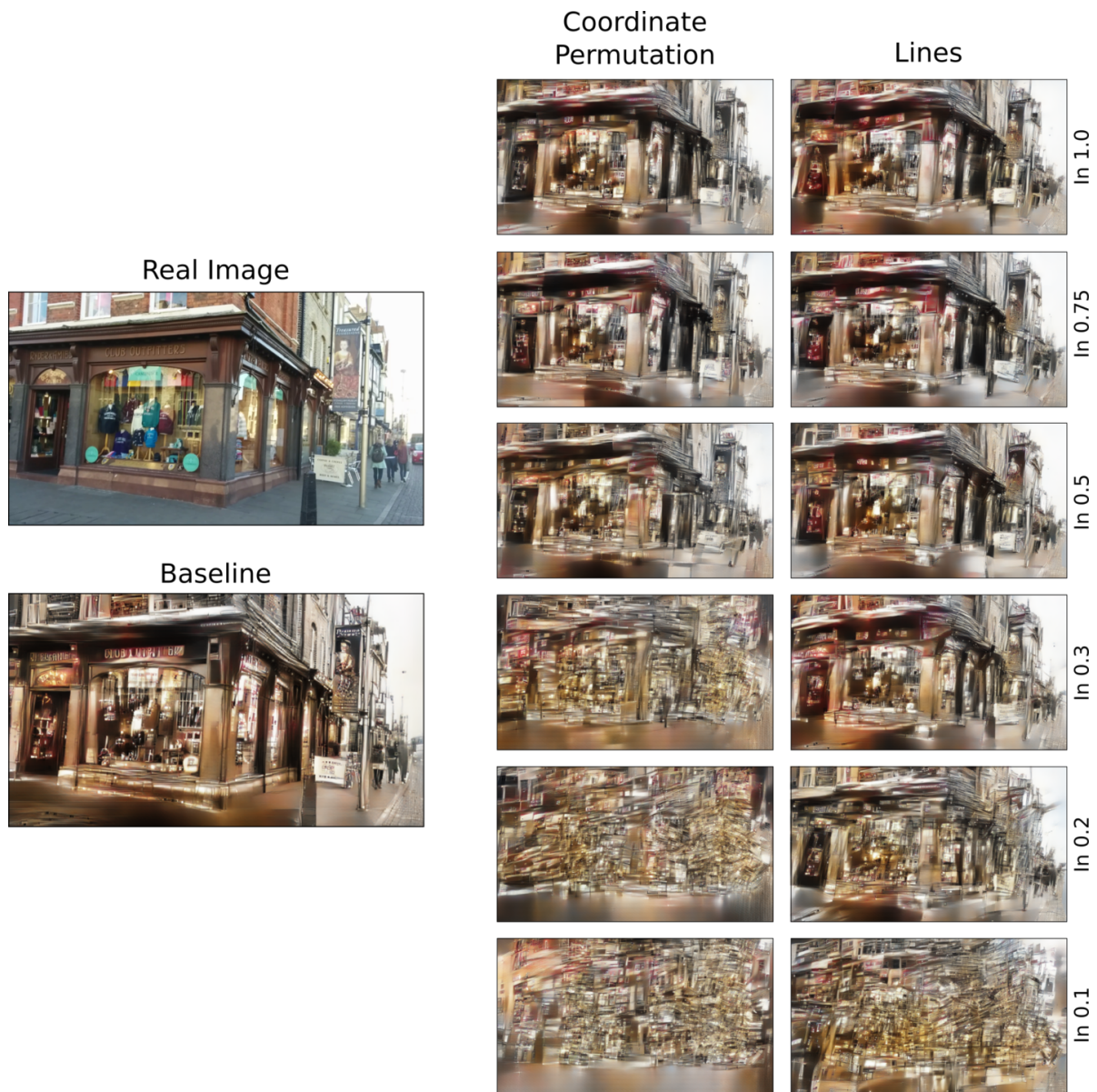


Figure 11. **Additional Qualitative Results - Cambridge [7] dataset, scene *Shop Facade*.** Images inverted from the original 2D SIFT [11] keypoints and the points recovered from the 2D obfuscations using neighborhood information with various levels of inlier ratios (In.).



Figure 12. **Additional Qualitative Results - 12scenes [20] dataset, scene *Office1-manolis*.** Images inverted from the SIFT [11] descriptors and original 2D keypoints and the points recovered from the 2D obfuscations using neighborhood information with various levels of inlier ratios (In.).

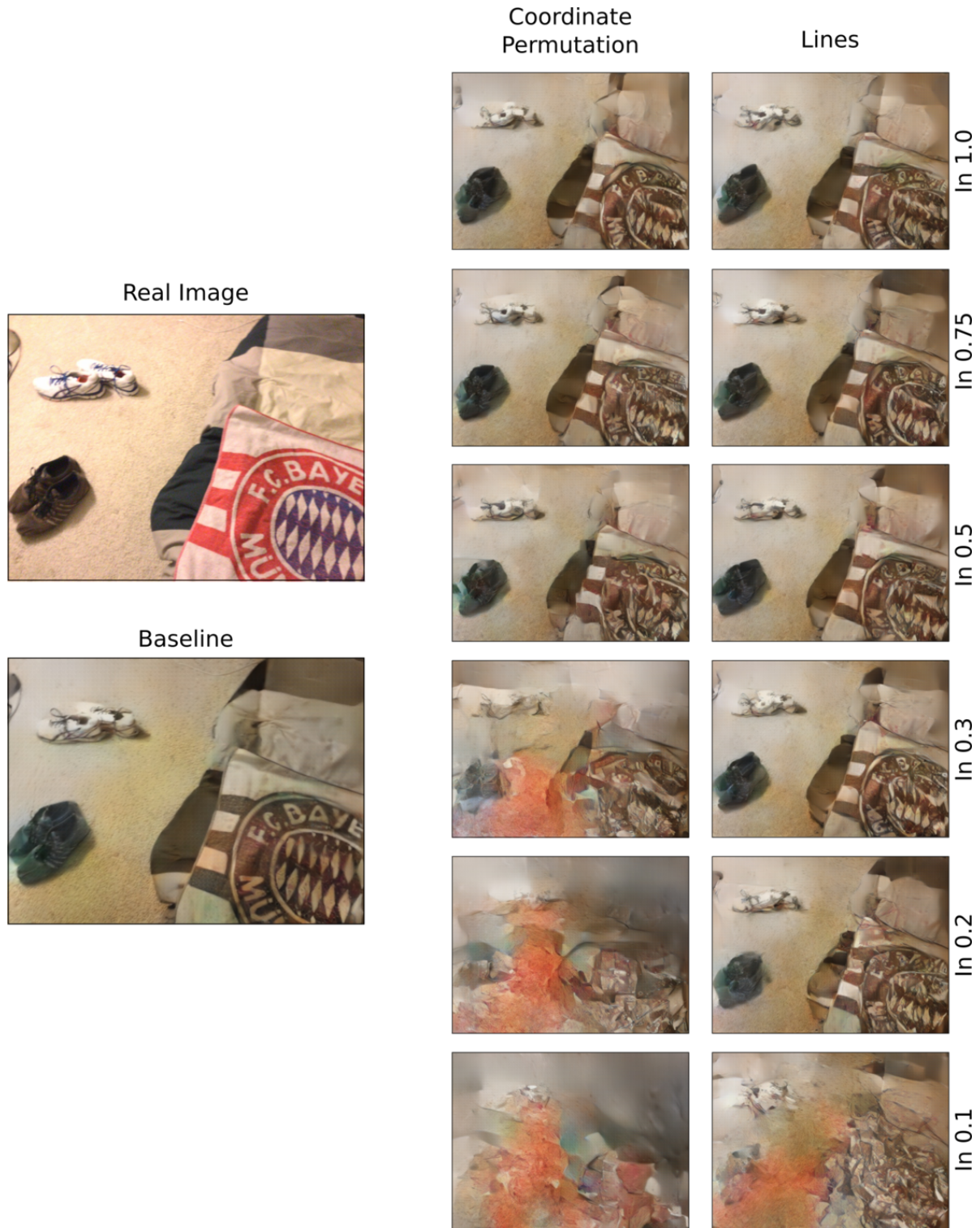


Figure 13. **Additional Qualitative Results - 12scenes [20] dataset, scene *Apt2-bed*.** Images inverted from the SIFT [11] descriptors and original 2D keypoints and the points recovered from the 2D obfuscations using neighborhood information with various levels of inlier ratios (In.).

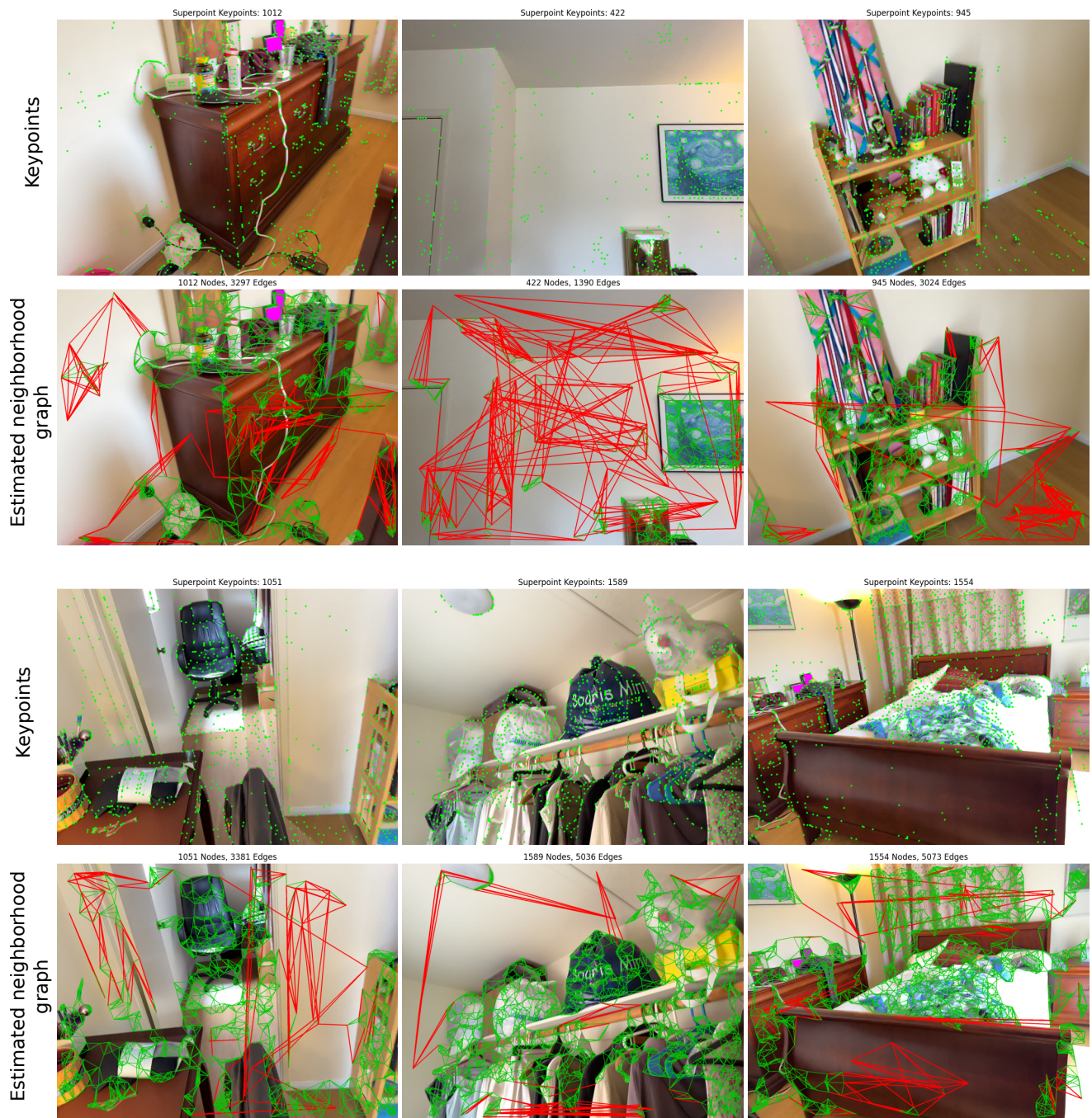


Figure 14. **Neighborhood estimated from SuperPoint [4] descriptors:** Images from scene *0a76e0647* from the ScanNet++ [22] showing detected SuperPoint [4] keypoints and the neighborhood graph estimated using our network described in Sec. 5 of the main paper. Top-5 neighbors for each point have been plotted with edges colored **green** if the points are closer than a threshold and **red** otherwise.

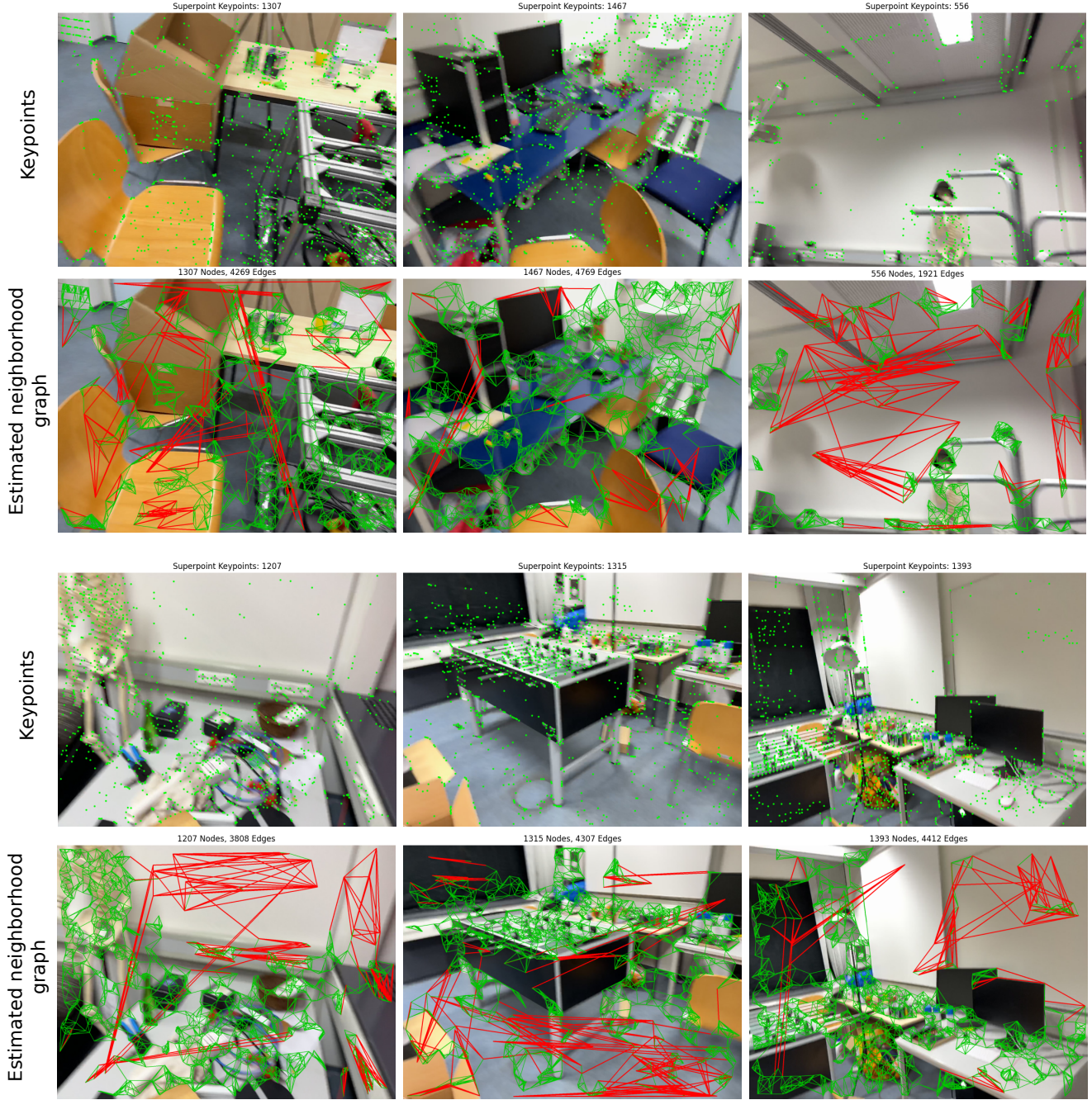


Figure 15. **Neighborhood estimated from SuperPoint [4] descriptors:** Images from scene *036bce3393* from the ScanNet++ [22] showing detected SuperPoint [4] keypoints and the neighborhood graph estimated using our network described in Sec. 5 of the main paper. Top-5 neighbors for each point have been plotted with edges colored **green** if the points are closer than a threshold and **red** otherwise.