Privacy-Preserving Structureless Visual Localization via Image Obfuscation

Supplementary Material

Vojtech Panek[1,2] Patrik Beliansky[1,2] Zuzana Kukelova[3] Torsten Sattler[1]
[1] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague
[2] Faculty of Electrical Engineering, Czech Technical University in Prague
[3] Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
{vojtech.panek,patrik.beliansky,torsten.sattler}@cvut.cz kukelzuz@fel.cvut.cz

In this supplementary material, we present the experiments and details that did not fit into the main paper. Sec. A contains a summary and visualizations of the used obfuscation methods. Sec. B discusses in more detail the privacy preservation properties of these methods, as well as the privacy preservation properties of state-of-the-art methods. Sec. C contains more details on the datasets used for the evaluation. Sec. D describes a pose refinement approach usable for segmentation-based obfuscation methods. Sec. E describes the implementation details for all the obfuscations and the localization pipelines. Sec. F presents an extended experimental evaluation from the main paper and an evaluation of the segment-based pose refinement method.

A. Used obfuscation methods

In this section, we summarize the tested obfuscation methods and show their examples.

A.1. Blurring and pixelization

Fig. 1 contains examples of the `blur` and `pixelization` obfuscations.

Figure 1. Gaussian `blur` and `pixelization` examples. From left to right: The original image, blur with a Gaussian kernel of size 81px, and a pixelization with 20x downsampling.

A.2. Selective anonymization

The selective anonymization methods first generate a binary mask based on semantic segmentation for a selected set of privacy-revealing classes. They then fill the corresponding areas in the original images using a selected method. More details on the segmentation model and the masked classes are presented in Sec. E. The infill for `easy-anon - single` is a single selected color (black) and `easy-anon - inpaint` applies inpainting [54] on each of the separate regions in the mask based on the pixels just outside the masked region. We show an example of the masks and anonymized images in Fig. 2.

Figure 2. Selective anonymization example. From top left to right: The original image, the segmentation of privacy-revealing classes, `easy-anon - single` and `easy-anon - inpaint`.

A.3. Segmentation

Fig. 3 contains a visual comparison of `Mask2Former - semantic` and `SAM1 - fine masks` on the Indoor-6 dataset [11]. The `Mask2Former` masks (Fig. 3 right) of indoor images are much more information-rich than those in outdoor scenes (see Fig 2 in the main paper), where they tend to segment mainly whole building outlines; still, the `SAM` masks (Fig. 3 middle) contain significantly more information.

Figure 3. Comparison of segmentation masks on the Indoor-6 dataset. Left to right: `original image`, `SAM1 - fine masks` segmentation and `Mask2Former - semantic` segmentation (using ADE20k classes).

A.4. Edge extraction

Apart from the Canny edge detector (`Canny`), which is applied directly to the input RGB photos, we tested an approach that first estimates monocular depth maps generated with the Metric3D [24, 62] monocular depth predictor and then extracts edges with the Canny detector on top of the depth maps (`Metric3D` → `Canny`). We also tested DiffusionEdge [61] (`DiffusionEdge`) as a representative of learned edge detectors. A visual comparison of edge-based obfuscations can be seen in Fig. 4.

Figure 4. Examples of the used edge extraction methods. From left to right: Canny edge detector used directly on the image, Canny edge detector applied on a Metric3D depth map, and an edge map extracted with DiffusionEdge.

B. Discussion on privacy

In this section, we discuss the privacy preservation properties of different methods. Traditional feature-based localization methods, which use 2D or 3D points with associated descriptors, are highly susceptible to inversion attacks [13, 45, 58], which can generate images containing privacy-revealing details. If an unobfuscated 3D point cloud is used [42, 43, 56, 68], it also directly reveals (sparse) information about scene geometry and, depending on the density of the points, the contents of the scene might be recognizable.

State-of-the-art obfuscation methods. Geometry obfuscation methods [18–20, 31, 38, 41, 51, 52] try to make feature inversion impossible by hiding the original positions of the points. However, the original point positions can be recovered for these geometric obfuscation methods [7, 8].[1] In turn, this makes inversion attacks applicable again. Given that the feature descriptors contain a lot of information, it is thus possible to recover texture details, personally identifying information such as faces, gender, text documents, *etc*., as well as concrete types of objects (*e.g*., the maker and model of a car, the content of a (potentially expensive) painting on the wall, *etc*.).

Descriptor obfuscation [15, 27, 28, 40, 42–44, 57, 68] focuses on making feature inversion more difficult by adjusting the design of descriptors. We are not aware of any attack that could overcome the recent work on descriptor obfuscation [44] and therefore we consider them to be privacy-preserving.

Blurring and pixelization. When using a large-enough kernel for `blur` or a large-enough `pixelization` downsampling factor, image details are no longer directly visible; however, the obfuscated image still preserves some of the original pixel information (convolved with a kernel, or downsampled). We do not consider such methods to be privacy-preserving, as they can be inverted back to the original image. For example, an attacker could exploit the fact that there will be multiple images of the same place to recover the original images using (single or multi-image) super-resolution [1, 2, 9, 12, 23, 32] or other methods [6, 22, 37, 53].

Selective obfuscation. The advantage of selective obfuscation is that the user can predefine objects that they consider privacy-revealing and these objects are completely removed from the images (up to the accuracy of the segmentation model). Note that `easy-anon - inpaint` uses only data outside of the masked area. Thus, the information from the masked area is completely lost (unlike in case of `blur` or `pixelization`) and trying to infer details of the original content in the masked areas is not possible. However, one can still identify the general object class based on its shape and surrounding context (such as the people or signs in Fig. 2). The shape of the masked region can also reveal information such as car type or gender of a person. However, if the masked object is not absolutely unique, the shape is not sufficient to identify details about the object. Masking out bounding boxes can help alleviate the issue of being able to identify the types of objects that were masked out. The layout of the rest of the scene, which is not masked out, is preserved in full detail.

One disadvantage of selective obfuscation is that the list of classes of privacy-sensitive objects, shapes, *etc*., must be known. At the same time, detectors for each of these classes need to be available (otherwise, these objects would need to be masked out manually). Training such detectors, *e.g*., for rare classes such as jewelery, can be a problem in itsef.

Segmentation-based obfuscation. The segmentation-based obfuscations used in our work follow the definition of privacy from [3, 42, 43]: In their definition, revealing the

[1] Given neighborhood information, *i.e*., information about which obfuscated points are close to each other in the original image, [8] show that the original point positions can be recovered for essentially all existing geometric obfuscation methods. This neighborhood information can be obtained, *e.g*., from the descriptors [8].

layout of the scene and the types of objects present is considered privacy-preserving as long as it is not possible to recover concrete details, *e.g.*, which painting is hanging on the wall, the maker and type of the TV, the titles of books on a bookshelf, *etc*.

[42, 43] advocate that the use of segmentations preserves privacy due to the observation that many different pixels can be mapped to the same segment label. This many-to-one mapping makes inversion attacks ill-posed. *E.g.*, given a segment corresponding to the head of a human, it is impossible to tell whether a person has blue eyes and blond hair or green eyes and dark hair. This introduces ambiguities in the inversion process, as shown in [3], since inversion attacks can infer pixel colors only on the basis of object shapes and the layout of the scene. Since each segmentation can be explained by a large space of possible images, the resulting recovered images are often unfaithful of the true scene content, even compared to the specialized geometry obfuscation methods [3] (see Figures 2 and 3 in [3]).

Segmentations only preserve the shapes of scene elements, while completely removing any information about color or texture. `Mask2former - semantic` also preserves the semantic class IDs in the segment coloring, while the other tested segmentation-based obfuscation methods use either random coloring of segments or render only the segment outlines. As for the selective anonymization, the shape of the segments can reveal some characteristics of the segmented object or person, but cannot uniquely identify them (as long as the shape is not unique to an object, person, *etc*.). The SAM-based segmentation detail granularity is tunable in a similar way as for edge extractors, which allows for easy adjustment of the amount of directly visible information (and the linked localization accuracy). The obfuscations using randomly colored segments are equivalent in privacy preservation to the methods using segment borders, as the random colors do not reveal any additional information, and all the information is contained within the shapes of the segments.

Edge extraction. Edge extraction, like segmentation-based methods, removes all color information. The level of directly visible information greatly depends on the edge detection sensitivity set by the user (*e.g.*, the size of kernels used by the Canny detector). Unlike a semantic segmentation map that captures only outlines of objects or their parts, an edge map generated with a sensitive edge detector can contain information about the texture of the object and therefore potentially reveal more information or make object shape reconstruction easier. The potential inversion attacks could then reconstruct the images up to the colors and low-gradient details.

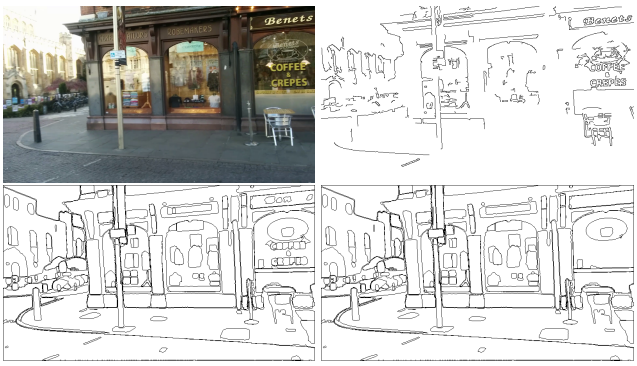

Figure 5. Comparison of `Canny` and `SAM1 - fine borders` on Cambridge Landmarks dataset. On the top row: `original image` and `Canny` edge map. The bottom row contains two maps generated with `SAM1 - fine borders`. The right one was generated with text filtration.

B.1. Server-side attack

So far, we have discussed privacy in relation to individual images. However, if an attacker gains access to the server, they can retrieve multiple images of the scene. The results of our method using local triangulation indicate that, in that case, an attacker might triangulate the 3D structure of the scene. This issue is common with feature-based methods, where the 3D structure may be readily available on the server or triangulated from 2D features. In the case of segmentation-based obfuscations, multiple images can reveal more silhouette details, which might identify an object with a unique silhouette or reveal a person's gender. Multiple blurred or pixelized images can be more effectively reconstructed using multi-image super-resolution.

C. Details on the evaluation datasets

This section contains details on the evaluated datasets.

Aachen Day-Night v1.1 [48, 49, 63] is an outdoor dataset that captures a part of the historical center of Aachen in Germany. The reference set contains over 6k day-time images captured over a longer time period. The query set contains over 1k day-time and night-time images.

Cambridge Landmarks [26] is the second outdoor dataset we used for the evaluation. It captures five historic landmarks in Cambridge, UK. It consists of frames extracted from continuous video streams. All images are day-time, and both queries and reference images were captured at approximately the same time. Some papers do not evaluate on the Great Court scene; therefore, in the main paper, we provide averages over all five scenes and over the four scenes excluding the Great Court one.

Indoor-6 [11] and **Remove-360** [29, 30] are the two indoor datasets we used for evaluation. Indoor-6 contains 6 multi-room scenes, from which five are from apartments and one

is from an office space. Remove-360 is a dataset originally targeting evaluating object removals from 3D Gaussian Splatting clouds. It captures multiple scenes where the training pass contains an object, which is not present in the testing pass. From all the scenes, we used only the four which were captured indoors, from which three capture an apartment room and the last one captures an office meeting room. The images in both datasets are frames from a continuous video stream.

D. Pose refinement based on segmentation masks

Extracted segmentation masks and the resulting local feature matches are often far from pixel-perfect. If we assume that the individual segments are usually slightly smaller or larger than the ground truth, we can obtain more stable keypoints by extracting their centroids. To establish matches between the segment centroids, we first need to find the corresponding segments between a pair of images. We determine which local feature keypoints fall into which segments and then pick the segment pairs with the highest IoU (Intersection over Union) of the matching local features.[2] Finally, we establish a single 2D-2D match between the centroid points of the two matched segments. Segment-level matches can be used directly for pose estimation or only for local optimization after pose estimation using standard local features.

E. Implementation details

For the `blur` obfuscation, we evaluate two Gaussian kernels, one of size $k_{\text{size}} = 41$ px ($\sigma = 6.5$ px) and the second of size $k_{\text{size}} = 81$ px ($\sigma = 12.5$ px).

The `pixelization` obfuscation is implemented as resizing the image to the selected fraction size ($1/10$ or $1/20$ in our experiments) followed by resizing to the original size using nearest-neighbor interpolation.

For anonymization with the easy-anon tool, we used a union of masks generated with the ResNet-101 and Swin-L Mask2Former models trained on the ADE20K dataset [66, 67]. The masked classes are:

- Animal
- Person
- Car
- Boat
- Bus
- Truck
- Plane
- Van
- Ship
- Motorbike
- Bicycle
- Painting
- Mirror
- Sign
- Book
- Computer
- TV
- Poster
- Screen
- CRT Screen
- Monitor
- Bulletin board
- Clock
- Flag

The `easy-anon - single` approach uses black color to fill the masked regions. The masked semantic labels include a number of privacy-revealing classes, such as people, vehicles, text boards, posters, or computer screens.

`Canny` uses the extractor implemented in OpenCV with the L1 gradient approximation. Both the `Canny` and `Metric3D` $\rightarrow$ `Canny` use a Gaussian kernel and Sobel kernels of size 3. The low and high thresholds were experimentally selected to remove small details while preserving features on the facades of the buildings. The original images are preprocessed using CLAHE [46] adaptive histogram equalization before the edge extraction to reduce the influence of lighting conditions. Both the `coarse` and `fine` `SAM1` variants use the ViT-H model, while the `coarse` `SAM2` variant uses the Hiera-B+ model, and the `fine` `SAM2` variant uses the Hiera-L model. For both `SAM1` and `SAM2`, the `coarse` variant infers the image in the 16×16 grid, while the `fine` variant uses the 32×32 grid and is inferred on image crops (by setting the 'crop_n_layers' parameter to 1). The masks for text filtration for `SAM`, described in the main paper, are created using a ResNet-50 model [34]. The inference points are first sampled in a grid (and potentially in the image crops), and then the ones falling into the regions with text are deleted. The remaining points are used to query the SAM model. The evaluated `Mask2Former` method uses the Swin-L panoptic segmentation model trained on the Mapillary Vistas dataset [39] for the evaluation on outdoor datasets and a model trained on ADE20K [66, 67] for the indoor datasets.

For retrieval, we use the EigenPlaces model based on ResNet-101, generating 2048-dimensional descriptors. During matching, a subset of RoMa matches is selected from a dense set based on their confidence. We found that for pose estimation, using the 1024 matches with the highest confidence gives the best results, while for establishing segment correspondences, using the 4096 best RoMa matches performs better.

To speed up the pose refinement using segmentation masks, we filter out all segments covering less than 100 px.

[2] The IoU of the matches between a segment pair is computed as the number of the matches over the number of features in both segments.

To find the corresponding segments, we use the 2D keypoints originating from the local feature matching. For every segment, we find the keypoints which fall into its mask dilated by 5 px (so that the keypoints on the segment borders are also taken into account).

F. Experiments

Extension of experiments from the main paper. Tabs. 1, 2, and 5 contain the results of experiments presented in the main paper, extended by the obfuscation methods which did not fit into the original tables. The additional experiments confirm the observations from the main paper. The selective obfuscation with `easy-anon` performs very similarly to the original images as it masks mainly dynamic objects. The infill type (`single` and `inpaint`) does not influence the extraction of local features outside of the masked regions, and therefore it does not matter which one is used for visual localization. The `SAM` segmentations outperform `Mask2Former`. The best results are achieved with the combination of `fine` segmentations, containing more visual information and `borders` rendering, which is preferred by the local feature matchers. `Canny` performs the best among the edge-extraction approaches.

Tab. 7 presents the full local feature matching ablation on the Remove360 dataset [29, 30]. We can see more often that MASt3R [33] surpasses RoMa [16] on the indoor scenes, *e.g.*, when matching two `Canny` edge maps. However, RoMa still offers more stable localization results without significant performance drops, which can be seen for MASt3R, *e.g.*, in case of matching between `original images` and `SAM1 fine masks`.

Using segment centroids for pose refinement. This section presents the evaluation of the pose refinement using 2D-2D matches between segment centroids (as defined in D). We tested two approaches for selecting the center of the segment. The first one simply computes the average of the coordinates of all the pixels belonging to the segment. The second fits a quadrilateral to the segment shape and computes the center as the intersection of the quadrilateral diagonals. Note that the quadrilateral fitting to each of the segments has high computational demands and results in a significant slowdown of the refinement. As both methods give very similar results, we decided to continue only with the more efficient of the two methods, and so the following experiments are using the simple coordinate averaging approach. The numerical comparison of the two methods is in Tab. 3.

The ablation on the segment-based pose refinement is presented in Tab. 4. We can see that the refinement does not necessarily bring an advantage to the localization pipeline, as the results with and without refinement are, on average, very similar. However, refinement can help for the finest pose error threshold for the nighttime queries.

SfM reconstructions using obfuscated images To experiment with "end-to-end" mapping and localization, we tried reconstructing the scenes using obfuscated images. While dense features provided better localization results, using them would have required developing a custom SfM pipeline to take into account that they do not provide repeatable keypoints. We thus only used sparse features - SuperPoint [10] and ALIKED [64, 65] - in combination with the LightGlue matcher [35] in order to use existing pipelines without modifications. To generate image pairs, we used exhaustive matching for Cambridge Landmarks [26] dataset and image retrieval (using CosPlace [4] with the ResNet101 [21] backbone, and Faiss [14]) of the 100 most relevant images for Indoor6 [11] dataset. After extracting features and matching images, we reconstructed the scenes using COLMAP [50]. For all reconstructions, we set the maximal number of features to 4096, and the minimal number of matches to 15. We used a single camera model for Cambridge Landmarks [26] dataset and distinct camera models for the other datasets.

In this setting, we observed failures when reconstructing certain scenes even from the original images. For example, in the St Mary's Church scene from Cambridge Landmarks [26], the reconstruction using the original images and ALIKED [64, 65] features produced a collapse of one side of the cathedral onto the other (see Figure 6). We noticed the same failure when reconstructing from the obfuscated images as well, using both SuperPoint [10] and ALIKED [64, 65] features. We also encountered reconstruction difficulties with the original images for the Great Court scene from Cambridge Landmarks [26] and scene3 and scene6 from the Indoor6 [11] dataset. The problem seems to be caused by incorrect matches between visually similar but physically distinct parts of the scene. Several works have addressed the problem of matching disambiguation [36, 59], and could be used to address these observed issues. In cases where the SfM reconstruction failed, the localization results were poor as well.

On the other hand, SfM on the obfuscated images performed very well on smaller or indoor scenes. For example, King's College, Old Hospital, and Shop Facade from the Cambridge Landmarks [26] dataset, as well as scene1 and scene4a from the Indoor6 [11] dataset. In some cases, these reconstructions even outperformed reconstructions from the original images. In most scenes, we observed better performance when using the SuperPoint [10] features. We show the localization experiments on the generated SfM poses on Cambridge Landmarks in the main paper and on the Indoor6 dataset in Tab. 6.

Image retrieval on obfuscated images All the other experiments extract global features and run image retrieval using non-obfuscated images. We assume that, because they are constructed using pooling, the global feature vectors repre-

	method	Great Court			King's College			Old Hospital			Shop Façade			St Mary's Church		
		MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]
E5+1	original images	0.29	0.13	43.2	0.19	0.30	60.6	0.24	0.48	51.6	0.06	0.28	95.1	0.10	0.31	86.8
	blur - 41 px	0.41	0.24	32.6	0.21	0.31	56.3	0.29	0.51	45.1	0.07	0.31	95.1	0.13	0.36	81.1
	blur - 81 px	1.12	0.94	7.4	0.43	0.76	22.2	0.52	1.06	17.0	0.19	0.66	68.0	0.44	1.34	24.5
	pixelization - 10x	0.65	0.39	22.5	0.37	0.63	31.5	0.61	1.11	24.2	0.10	0.42	87.4	0.15	0.50	75.7
	pixelization - 20x	6.82	5.59	1.1	1.87	3.08	1.5	2.17	3.46	1.1	0.53	2.55	15.5	1.13	3.48	3.4
	easy-anon - single	0.30	0.13	43.0	0.20	0.29	60.6	0.25	0.46	50.0	0.07	0.26	95.1	0.10	0.30	87.0
	easy-anon - inpaint	0.29	0.13	43.4	0.20	0.31	60.1	0.25	0.47	50.0	0.07	0.27	95.1	0.10	0.29	86.2
	Canny	0.42	0.21	30.9	0.19	0.28	61.2	0.25	0.51	50.0	0.06	0.22	94.2	0.11	0.30	82.6
	Metric3D → Canny	0.51	0.26	23.4	0.25	0.37	50.1	0.50	0.98	26.4	0.25	0.82	49.5	0.45	1.27	33.0
	DiffusionEdge	0.61	0.32	23.9	0.26	0.40	46.9	0.48	0.83	28.6	0.15	0.69	80.6	0.26	0.82	48.1
	SAM2 - coarse masks	0.50	0.26	24.1	0.24	0.38	53.1	0.45	0.82	28.6	0.10	0.42	84.5	0.23	0.74	54.0
	SAM2 - coarse borders	0.51	0.25	24.9	0.23	0.33	55.1	0.43	0.76	34.6	0.10	0.43	91.3	0.19	0.60	62.5
	SAM2 - fine masks	0.42	0.20	30.5	0.21	0.31	59.2	0.26	0.54	48.4	0.07	0.36	94.2	0.13	0.40	76.2
	SAM2 - fine borders	0.38	0.17	35.1	0.20	0.30	58.6	0.26	0.49	47.3	0.07	0.33	94.2	0.12	0.36	82.5
	SAM1 - coarse masks	0.41	0.21	30.8	0.21	0.32	57.1	0.44	0.82	32.4	0.08	0.35	92.2	0.16	0.56	69.8
	SAM1 - coarse borders	0.37	0.18	34.7	0.20	0.30	58.6	0.38	0.66	39.6	0.09	0.34	93.2	0.13	0.39	77.9
	SAM1 - fine masks	0.39	0.19	32.9	0.19	0.28	60.9	0.29	0.58	41.2	0.06	0.28	93.2	0.13	0.41	80.4
	SAM1 - fine borders	0.35	0.16	37.1	0.19	0.27	62.1	0.26	0.46	48.9	0.07	0.33	94.2	0.10	0.34	84.0
	Mask2Former - semantic	0.41	0.21	28.8	0.23	0.35	53.9	0.98	1.42	6.6	0.15	0.53	73.8	0.35	1.04	37.4
	Mask2Former - random	0.63	0.32	17.9	0.29	0.40	43.7	2.40	3.80	1.6	0.22	0.73	60.2	0.80	2.24	22.3
	Mask2Former - borders	0.45	0.21	26.8	0.22	0.33	55.7	1.40	2.26	4.9	0.13	0.50	69.9	0.34	0.93	39.4
LT	original images	0.29	0.13	43.7	0.20	0.30	61.2	0.24	0.50	50.5	0.06	0.27	95.1	0.10	0.30	86.8
	blur - 41 px	0.39	0.23	33.0	0.20	0.30	57.1	0.26	0.50	45.6	0.07	0.31	95.1	0.13	0.38	80.6
	blur - 81 px	1.20	1.01	6.7	0.43	0.76	23.6	0.54	1.10	16.5	0.18	0.60	70.9	0.45	1.32	24.5
	pixelization - 10x	0.62	0.40	20.8	0.40	0.65	30.6	0.58	1.01	28.6	0.10	0.39	89.3	0.16	0.49	73.8
	pixelization - 20x	6.61	5.43	2.0	1.94	3.12	2.0	2.22	3.47	1.6	0.64	2.90	15.5	1.06	3.50	5.7
	easy-anon - single	0.29	0.13	43.3	0.20	0.30	59.8	0.24	0.47	51.1	0.07	0.28	95.1	0.10	0.31	86.4
	easy-anon - inpaint	0.29	0.13	42.8	0.19	0.30	60.9	0.25	0.47	50.0	0.07	0.26	94.2	0.10	0.31	86.6
	Canny	0.41	0.20	32.0	0.19	0.29	60.9	0.25	0.47	51.6	0.06	0.25	94.2	0.11	0.30	82.6
	Metric3D → Canny	0.51	0.28	25.4	0.25	0.37	51.9	0.61	1.11	25.3	0.21	0.74	55.3	0.42	1.13	35.1
	DiffusionEdge	0.60	0.33	21.6	0.26	0.41	47.8	0.44	0.79	30.2	0.14	0.61	77.7	0.25	0.81	50.0
	SAM2 - coarse masks	0.52	0.27	23.0	0.24	0.36	51.9	0.45	0.90	31.3	0.10	0.43	87.4	0.25	0.72	50.6
	SAM2 - coarse borders	0.51	0.26	24.7	0.22	0.33	53.9	0.39	0.69	36.3	0.10	0.41	88.3	0.19	0.60	60.6
	SAM2 - fine masks	0.44	0.21	30.3	0.22	0.31	58.9	0.26	0.56	47.3	0.07	0.36	94.2	0.13	0.42	76.0
	SAM2 - fine borders	0.38	0.17	35.3	0.20	0.30	58.0	0.26	0.47	48.4	0.07	0.31	94.2	0.12	0.36	82.1
	SAM1 - coarse masks	0.41	0.22	30.7	0.22	0.32	57.1	0.48	0.83	31.9	0.08	0.35	93.2	0.17	0.55	70.0
	SAM1 - coarse borders	0.38	0.20	34.6	0.20	0.30	58.3	0.36	0.63	39.6	0.08	0.32	94.2	0.13	0.41	76.6
	SAM1 - fine masks	0.40	0.19	33.6	0.21	0.29	60.9	0.30	0.58	42.9	0.06	0.30	95.1	0.13	0.39	78.7
	SAM1 - fine borders	0.35	0.15	36.8	0.19	0.27	61.8	0.26	0.45	48.4	0.07	0.33	92.2	0.10	0.34	84.5
	Mask2Former - semantic	0.41	0.21	27.2	0.23	0.34	53.9	1.16	1.83	4.9	0.15	0.52	73.8	0.33	0.85	38.9
	Mask2Former - random	0.64	0.34	18.9	0.28	0.40	44.0	2.63	4.00	3.8	0.18	0.65	63.1	0.77	2.37	23.0
	Mask2Former - borders	0.45	0.22	26.2	0.23	0.33	54.8	1.35	1.87	6.6	0.14	0.49	72.8	0.31	0.91	41.9

Table 1. Localization results on Cambridge Landmarks [26] using the top-20 reference images retrieved with EigenPlaces [5], RoMa [16] feature matching, and the E5+1 pose solver. We report median position (MPE) and orientation (MOE) errors (smaller is better) and recall (rec.) at 25 cm, 2° pose error (higher is better).

sent only general information about the scene (such as room type) rather than visual details from the images. As such, sending the global features to the localization server along with the obfuscated images does not significantly increase privacy risk. In this experiment, we test whether global features extracted from the original images can be replaced by those extracted from the obfuscated images, in which case global feature extraction could be performed directly on the server.

The results presented in Tab. 8 show that, for most methods, the localization recall significantly drops when using an off-the-shelf retrieval method (EigenPlaces [5]) based on obfuscated images. There are a few obvious exceptions, such as the `easy-anon` methods, where obfuscated images do not differ significantly from the originals and retrieval performance is similar. In summary, EigenPlaces is not robust to most image obfuscations, and retrieval based on original images is necessary to prevent a significant drop in accuracy.

Runtime and bandwidth We present a runtime analysis of our approach in Table 10. Localization using images obfuscated with `SAM1 - fine borders` took longer com-

obfuscation method	Bedroom Table			Living Room (1) Pillows			Living Room (2) Sofa			Office Chairs		
	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]
original images	0.01	0.42	93.5	0.01	0.37	94.6	0.01	0.22	98.5	0.01	0.26	89.4
blur - 41 px	0.04	0.93	60.0	0.02	0.55	83.3	0.02	0.51	75.4	0.03	0.51	70.0
blur - 81 px	0.10	2.20	19.2	0.05	1.37	53.8	0.06	1.55	45.8	0.09	1.61	26.7
pixelization - 10x	0.04	0.99	62.3	0.02	0.72	84.6	0.03	0.75	68.2	0.05	0.90	52.1
pixelization - 20x	0.16	3.81	9.2	0.09	2.45	26.2	0.16	3.74	17.0	0.45	6.42	3.3
easy-anon - single	0.01	0.45	94.2	0.01	0.36	94.6	0.01	0.23	98.5	0.02	0.27	88.5
easy-anon - inpaint	0.02	0.42	94.2	0.01	0.37	94.6	0.01	0.23	98.5	0.01	0.26	89.4
Canny	0.06	1.54	43.5	0.04	1.05	57.5	0.02	0.47	70.1	0.03	0.51	61.2
Metric3D → Canny	0.39	7.85	8.1	0.23	6.96	23.1	0.09	1.74	41.3	0.23	3.59	13.0
DiffusionEdge	0.17	3.94	21.5	0.06	1.90	45.2	0.04	1.05	53.4	0.14	1.96	20.9
SAM2 - coarse masks	0.06	1.47	46.2	0.03	0.79	66.1	0.03	0.81	56.8	0.06	0.93	47.6
SAM2 - coarse borders	0.05	1.14	52.7	0.03	0.66	70.1	0.03	0.69	60.2	0.06	0.93	48.2
SAM2 - fine masks	0.04	0.92	60.0	0.02	0.61	73.3	0.02	0.45	68.2	0.03	0.60	61.2
SAM2 - fine borders	0.03	0.84	65.4	0.02	0.54	74.2	0.02	0.42	74.2	0.03	0.57	62.4
SAM1 - coarse masks	0.04	1.00	58.8	0.02	0.63	71.9	0.02	0.48	70.1	0.03	0.59	57.9
SAM1 - coarse borders	0.03	0.87	61.2	0.02	0.50	76.0	0.02	0.48	75.0	0.03	0.55	65.8
SAM1 - fine masks	0.04	0.88	57.3	0.02	0.60	72.4	0.02	0.49	67.4	0.04	0.63	58.2
SAM1 - fine borders	0.03	0.75	62.7	0.02	0.56	78.7	0.02	0.42	72.3	0.03	0.59	58.5
Mask2Former - semantic	0.03	0.79	61.2	0.02	0.56	75.1	0.02	0.41	74.2	0.03	0.59	62.4
Mask2Former - random	0.03	0.82	65.4	0.02	0.54	77.4	0.02	0.35	74.2	0.02	0.45	67.6
Mask2Former - borders	0.04	0.87	58.5	0.02	0.61	74.2	0.02	0.46	72.0	0.03	0.56	60.6

Table 2. Localization results on the Remove 360 [29, 30] dataset, using the top-20 reference images retrieved with EigenPlaces [5], RoMa [16] feature matching, and the E5+1 pose solver. We reporting median position (MPE) and orientation (MOE) errors (smaller is better) and recall (rec.) at 5 cm, 5° pose error (higher is better).

	obfuscation	centroid	day	night
SAM1	fine masks	coord. avg.	63.0 / 80.6 / 94.1	49.7 / 73.8 / 96.3
	fine masks	quad. center	63.1 / 80.6 / 94.1	48.7 / 73.8 / 95.8
	fine borders	coord. avg.	68.6 / 84.1 / 95.3	54.5 / 82.2 / 95.8
	fine borders	quad. center	68.9 / 84.1 / 95.3	54.5 / 81.7 / 95.8

Table 3. Ablation on segment centroid computation method used for pose refinement. Showing results on Aachen Day-Night v1.1 [48, 49, 63], using the top-20 retrieved reference images with EigenPlaces [5] and RoMa [16] feature matches for pose estimation with E5+1 and segment centroid matches for pose refinement. We reporting localization recalls (higher is better) at the pose error thresholds of (0.25 m, 2°) / (0.5 m, 5°) / (5 m, 10°).

	obfuscation	refined	day	night
	original images	✗	77.9 / 90.5 / 98.2	64.9 / 88.5 / 98.4
SAM2	coarse masks	✗	28.6 / 46.8 / 77.7	17.8 / 34.0 / 70.2
		✓	31.3 / 47.3 / 75.0	18.3 / 36.6 / 67.0
	coarse borders	✗	36.2 / 50.1 / 78.2	20.9 / 36.1 / 70.7
		✓	35.3 / 51.1 / 76.9	21.5 / 37.7 / 67.5
	fine masks	✗	52.1 / 69.8 / 90.5	30.9 / 56.0 / 85.9
		✓	52.2 / 71.5 / 90.8	34.0 / 57.1 / 86.9
	fine borders	✗	59.2 / 74.5 / 91.4	36.6 / 63.4 / 89.0
		✓	57.2 / 74.5 / 91.0	39.8 / 64.4 / 86.4
SAM1	coarse masks	✗	48.5 / 69.4 / 91.9	31.9 / 58.1 / 92.1
		✓	51.5 / 70.8 / 92.5	33.5 / 60.7 / 91.1
	coarse borders	✗	58.1 / 76.0 / 93.0	42.9 / 66.0 / 95.8
		✓	57.3 / 75.5 / 92.1	43.5 / 67.0 / 93.7
	fine masks	✗	63.6 / 80.5 / 94.9	42.4 / 73.8 / 96.9
		✓	64.2 / 81.6 / 95.3	49.2 / 77.0 / 96.3
	fine borders	✗	71.7 / 83.1 / 96.1	53.9 / 80.1 / 97.4
		✓	70.1 / 84.2 / 95.4	53.4 / 82.2 / 96.9

Table 4. Localization results on Aachen Day-Night v1.1 [48, 49, 63] using the top-20 retrieved reference images with EigenPlaces [5] and RoMa [16] feature matches for pose estimation with E5+1 and segment centroid matches for pose refinement. We reporting localization recalls (higher is better) at the pose error thresholds of (0.25 m, 2°) / (0.5 m, 5°) / (5 m, 10°).

pared to the original images, probably due to a significantly higher number of false matches, resulting in more RANSAC iterations. The obfuscation stage in our approach necessarily brings some computational overhead, see Table 11. We observed that obfuscating images with SAM1 and SAM2 was the slowest, which is expected given the large size of the SAM models. This could be addressed by using smaller, distilled versions of the models [60]. Coarse segmentations required a similar processing time for both SAM1 and SAM2, while fine segmentations were significantly slower with SAM2. Other privacy-preserving approaches, such as the geometric obfuscation methods [18–20, 31, 38, 51, 52] also have their own computational overhead due to the complexity of their pose solvers (*e.g.*, generalized 6pt solver or P6LP), which sample six points/lines rather than using the more efficient P3P solver.

The difference in bandwidth requirements (to transfer

	method	scene1			scene2a			scene3			scene4a			scene5			scene6		
		MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]	MPE [m]	MOE [°]	rec. [%]
E5+1	original images	0.01	0.16	92.9	0.01	0.11	94.2	0.01	0.12	97.1	0.01	0.17	96.8	0.01	0.19	92.7	0.01	0.12	95.7
	blur - 41 px	0.02	0.27	84.7	0.02	0.19	85.2	0.01	0.21	92.1	0.01	0.31	91.8	0.03	0.42	75.0	0.01	0.19	93.5
	blur - 81 px	0.04	0.70	63.1	0.05	0.53	51.4	0.03	0.63	69.8	0.03	0.81	63.3	0.07	1.08	36.1	0.02	0.42	82.4
	pixelization - 10x	0.02	0.31	87.4	0.02	0.26	76.3	0.01	0.27	90.5	0.02	0.41	84.8	0.03	0.50	73.3	0.01	0.23	91.0
	pixelization - 20x	0.07	1.31	37.7	0.11	1.02	21.8	0.06	1.11	46.0	0.06	1.34	44.3	0.10	1.57	17.9	0.04	0.94	61.3
	easy-anon - single	0.01	0.18	92.4	0.01	0.12	93.0	0.01	0.13	96.2	0.01	0.21	93.0	0.01	0.20	87.5	0.01	0.13	94.1
	easy-anon - inpaint	0.01	0.17	92.0	0.01	0.12	91.4	0.01	0.13	96.5	0.01	0.20	96.8	0.01	0.20	89.9	0.01	0.12	95.0
	Canny	1.15	24.55	9.4	0.10	1.00	42.0	0.63	11.94	26.0	1.08	22.10	17.1	1.55	23.75	8.0	0.53	13.07	27.2
	Metric3D → Canny	0.44	8.08	15.3	0.26	3.26	17.1	0.82	15.36	12.1	0.58	11.96	12.0	1.04	13.92	3.5	0.35	8.42	22.6
	DiffusionEdge	0.05	0.88	52.2	0.05	0.55	52.9	0.04	0.77	58.1	0.06	1.29	45.6	0.08	1.22	38.9	0.03	0.67	65.9
	SAM1 - fine masks	0.02	0.42	75.8	0.02	0.24	79.0	0.02	0.32	77.8	0.02	0.51	72.2	0.03	0.41	68.4	0.01	0.28	80.8
	SAM1 - fine borders	0.02	0.34	78.0	0.02	0.21	79.8	0.01	0.27	85.7	0.02	0.46	74.1	0.02	0.41	71.9	0.01	0.21	88.2
	Mask2Former - semantic	0.09	1.58	29.8	0.06	0.69	44.4	0.08	1.40	39.7	0.18	3.60	18.4	0.10	1.57	28.3	0.16	2.78	31.6
	Mask2Former - random	0.11	1.91	30.8	0.07	0.70	41.6	0.09	1.64	38.1	0.25	5.73	16.5	0.11	1.68	26.9	0.14	3.34	33.1
	Mask2Former - borders	0.08	1.35	37.2	0.05	0.57	49.0	0.07	1.33	42.2	0.26	5.15	15.8	0.09	1.39	34.2	0.12	2.27	39.9
LT (Local Triangulation)	original images	0.01	0.27	84.9	0.01	0.11	93.4	0.01	0.20	90.8	0.02	0.41	79.1	0.03	0.41	75.7	0.01	0.23	91.0
	blur - 41 px	0.03	0.47	73.1	0.02	0.19	83.7	0.02	0.40	81.0	0.03	0.60	72.8	0.05	0.81	52.4	0.01	0.34	85.1
	blur - 81 px	0.06	1.03	47.4	0.05	0.50	54.9	0.05	1.07	49.8	0.06	1.50	40.5	0.10	1.69	21.0	0.03	0.66	68.7
	pixelization - 10x	0.03	0.56	68.7	0.03	0.27	79.0	0.02	0.49	69.2	0.04	0.90	56.3	0.06	0.90	46.7	0.02	0.38	80.8
	pixelization - 20x	0.10	1.92	18.3	0.11	1.09	21.4	0.11	2.37	26.0	0.12	2.89	20.3	0.18	2.91	7.8	0.07	1.58	40.6
	easy-anon - single	0.02	0.30	82.7	0.01	0.11	93.4	0.01	0.21	90.2	0.02	0.52	77.8	0.03	0.45	73.6	0.01	0.23	90.1
	easy-anon - inpaint	0.02	0.29	83.2	0.01	0.12	91.1	0.01	0.21	88.9	0.02	0.50	77.8	0.03	0.44	72.9	0.01	0.22	89.5
	Canny	0.90	21.43	8.3	0.10	0.92	44.0	0.42	8.98	23.8	0.50	12.20	15.2	1.34	24.60	5.2	0.29	6.51	31.9
	Metric3D → Canny	0.27	5.47	16.4	0.21	2.53	17.9	0.63	12.00	13.0	0.58	11.69	10.8	0.99	16.18	2.6	0.26	5.59	21.7
	DiffusionEdge	0.05	0.93	48.6	0.05	0.51	53.3	0.05	0.97	50.2	0.08	1.66	38.6	0.07	1.26	39.6	0.03	0.78	63.2
	SAM1 - fine masks	0.03	0.58	67.6	0.02	0.25	77.8	0.03	0.59	63.8	0.04	0.85	62.0	0.04	0.68	54.5	0.02	0.41	75.2
	SAM1 - fine borders	0.03	0.50	70.5	0.02	0.22	79.4	0.02	0.40	75.2	0.03	0.66	64.6	0.04	0.60	60.8	0.01	0.31	80.2
	Mask2Former - semantic	0.10	1.62	29.2	0.07	0.74	42.0	0.12	2.13	32.4	0.32	6.42	10.1	0.12	2.03	25.9	0.17	3.18	34.4
	Mask2Former - random	0.11	1.91	27.7	0.07	0.76	40.5	0.12	2.09	33.3	0.38	9.39	8.9	0.16	2.54	24.1	0.20	4.16	30.3
	Mask2Former - borders	0.07	1.33	37.0	0.05	0.53	50.2	0.09	1.69	38.4	0.28	5.95	13.3	0.11	1.72	28.8	0.11	2.11	39.0

Table 5. Localization results on Indoor-6 [11] using the top-20 reference images retrieved with EigenPlaces [5], RoMa [16] feature matching, and the E5+1 pose solver as well as local triangulation. We report median position (MPE) and orientation (MOE) errors (smaller is better) and recall (rec.) at 5 cm, 5° pose error (higher is better).

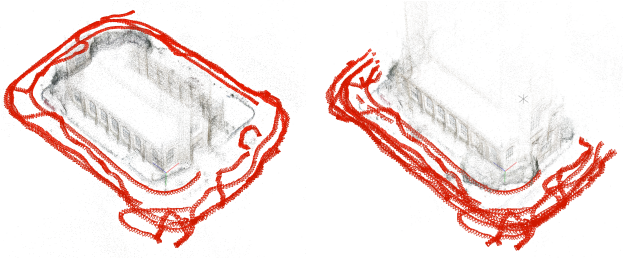

Figure 6. Reconstructions of the St Mary's Church scene from Cambridge Landmarks [26] from original images using SuperPoint [10] features (left) and ALIKED [64, 65] features (right). We can see that in the right image, one side of the cathedral collapsed onto the other.

files between user and the localization server) depends on implementation details. Feature-based methods transfer detected keypoints and descriptors, which can result in a substantial amount of data due to the size of the descriptors. Structureless methods transfer images, whose data sizes depend on the used encoding and image size. We show a brief comparison in Tab. 12. If the system were to extract and transfer 1000 features, the data size of the resulting features is about the same as the size of the images. An average SAM border mask encoded as a bilevel PNG is equivalent in size to 250 ALIKED or 126 SuperPoint features.

Comparing different feature matching methods. We compared the performance of pose estimation with E5+1 with different feature matching methods. We tested RoMa [16], RoMa v2 [17], Tiny RoMa [16], XFeat [47], and DISK [55] features in combination with the LightGlue matcher [35]. We observe that localization using images obfuscated with `SAM1 - fine borders` yields good results with RoMa [16] and RoMa v2 [17] matching, but drops significantly with other matching methods. For obfuscated images, we got the best results with RoMa [16], while for the `original images` RoMa v2 [17] performed better (see Table 9).

	method	ref. poses	scene1 MPE [m]	scene1 MOE [°]	scene1 rec. [%]	scene2a MPE [m]	scene2a MOE [°]	scene2a rec. [%]	scene3 MPE [m]	scene3 MOE [°]	scene3 rec. [%]	scene4a MPE [m]	scene4a MOE [°]	scene4a rec. [%]	scene5 MPE [m]	scene5 MOE [°]	scene5 rec. [%]	scene6 MPE [m]	scene6 MOE [°]	scene6 rec. [%]	average MPE [m]	average MOE [°]	average rec. [%]
E5+1	easy-anon - single	obf.	0.03	0.51	73.1	0.03	0.36	66.5	0.04	0.70	59.7	0.02	0.68	79.1	0.04	0.61	61.6	0.02	0.39	84.8	0.03	0.54	70.80
	SAM1 - fine borders	obf.	0.05	1.14	49.7	0.07	0.56	31.1	0.04	0.68	63.2	0.03	0.77	60.8	0.05	0.87	48.6	0.04	0.94	56	0.05	0.83	51.57
LT	easy-anon - single	obf.	0.04	0.56	64.2	0.04	0.35	65.4	0.05	0.82	55.2	0.05	1.14	51.3	0.05	0.88	48.8	0.02	0.45	78.3	0.04	0.70	60.5
	SAM1 - fine borders	obf.	0.06	1.34	38.3	0.07	0.58	30.0	0.05	1.02	50.2	0.06	1.37	43.7	0.07	1.16	36.6	0.05	1.10	44.9	0.06	1.09	40.6

Table 6. Localization results on Indoor-6 [11] using "end-to-end" pipeline with reference poses from SfM on obfuscated images (obf.). The experiments use top-10 reference images retrieved with EigenPlaces [5], SuperPoint local features and LightGlue matching, and pose estimation with E5+1 and local triangulation (LT). Reporting median position (MPE) and orientation (MOE) errors (smaller is better) and recall (rec.) at 5 cm, 5° pose error (higher is better).

query	ref.	matching	Bedroom Table MPE [m]	Bedroom Table MOE [°]	Bedroom Table rec. [%]	Living Room (1) Pillows MPE [m]	Living Room (1) Pillows MOE [°]	Living Room (1) Pillows rec. [%]	Living Room (2) Sofa MPE [m]	Living Room (2) Sofa MOE [°]	Living Room (2) Sofa rec. [%]	Office Chairs MPE [m]	Office Chairs MOE [°]	Office Chairs rec. [%]
orig. images	orig. images	RoMa	0.01	0.42	93.5	0.01	0.37	94.6	0.01	0.23	98.5	0.02	0.26	89.4
		SP+LG	0.02	0.40	91.2	0.01	0.37	94.1	0.01	0.27	92	0.02	0.32	82.1
		ALIKED+LG	0.02	0.44	86.2	0.01	0.40	90	0.01	0.27	89.4	0.02	0.36	73
		MASt3R	0.02	0.41	95.8	0.01	0.43	94.6	0.01	0.29	98.9	0.02	0.26	88.2
SAM1 f. masks	SAM1 f. masks	RoMa	0.03	0.79	61.2	0.02	0.56	75.1	0.02	0.41	74.2	0.03	0.60	62.4
		SP+LG	0.10	2.37	26.5	0.06	1.66	44.8	0.03	0.83	58	0.31	3.66	23.3
		ALIKED+LG	0.07	1.55	42.3	0.04	1.01	54.3	0.03	0.57	65.9	0.10	1.42	34.5
		MASt3R	0.03	0.79	70.8	0.03	0.81	65.6	0.03	0.61	64.8	0.04	0.71	53
SAM1 f. borders	SAM1 f. borders	RoMa	0.03	0.82	65.4	0.02	0.54	77.4	0.02	0.35	74.2	0.02	0.45	67.6
		SP+LG	0.04	0.97	59.2	0.02	0.62	73.3	0.02	0.48	72.3	0.03	0.65	58.8
		ALIKED+LG	0.04	1.12	53.5	0.03	0.93	61.1	0.02	0.54	66.3	0.06	0.99	45.8
		MASt3R	0.03	0.71	77.3	0.02	0.52	82.8	0.02	0.51	72	0.02	0.46	73.3
Canny	Canny	RoMa	0.06	1.54	43.5	0.04	1.05	57.5	0.02	0.47	70.1	0.03	0.51	61.2
		SP+LG	0.07	1.77	44.6	0.04	1.12	59.3	0.02	0.46	69.3	0.05	0.96	47
		ALIKED+LG	0.09	2.35	41.9	0.05	1.16	51.6	0.02	0.51	67.4	0.10	1.35	38.5
		MASt3R	0.03	0.80	65	0.03	0.63	74.7	0.02	0.56	71.6	0.02	0.43	78.8
original images	SAM1 f. masks	RoMa	0.07	1.60	39.6	0.03	0.76	61.1	0.02	0.53	68.6	0.06	0.90	46.1
		SP+LG	0.41	8.63	21.9	0.22	5.06	25.8	0.06	1.35	49.6	1.09	18.46	14.2
		ALIKED+LG	0.42	8.99	8.5	0.86	38.36	19.5	0.11	2.55	34.1	2.20	39.29	3.3
		MASt3R	0.21	3.50	21.2	0.47	7.59	14.5	0.58	5.54	8.7	1.05	8.12	4.8
	SAM1 f. borders	RoMa	0.06	1.56	43.8	0.04	0.85	59.7	0.02	0.53	69.7	0.06	1.07	46.7
		SP+LG	0.15	3.39	28.8	0.88	23.49	20.4	0.04	1.08	53	1.09	16.21	16.1
		ALIKED+LG	0.27	5.80	13.5	0.93	34.62	17.2	0.11	2.67	33.7	1.90	41.26	5.5
		MASt3R	0.07	1.49	37.3	0.35	7.94	13.6	0.17	3.52	23.1	0.19	2.13	21.5
	Canny	RoMa	0.63	12.89	23.8	0.22	5.58	31.7	0.02	0.61	61.4	0.09	1.43	38.2
		SP+LG	1.37	27.39	16.2	0.66	19.52	23.5	0.03	0.70	59.5	0.28	3.64	27.3
		ALIKED+LG	0.61	22.03	19.2	0.82	25.55	22.6	0.04	1.05	53.8	2.07	35.42	7.6
		MASt3R	0.07	1.65	38.1	0.15	3.48	14	0.06	1.25	43.6	0.05	0.87	49.4
SAM1 f. masks	original images	RoMa	0.10	2.58	33.5	0.05	1.32	49.8	0.02	0.54	66.3	0.05	0.90	48.8
		SP+LG	0.45	12.23	18.1	0.74	19.72	24.4	0.06	1.35	46.2	1.61	31.32	10.3
		ALIKED+LG	1.05	27.36	3.5	1.12	37.92	13.1	0.21	4.74	23.5	2.31	42.55	3.9
		MASt3R	1.05	27.98	11.9	1.81	51.04	4.1	1.94	56.34	4.5	2.84	62.64	3.9
SAM1 f. borders		RoMa	0.08	1.94	34.6	0.04	1.00	57.5	0.02	0.62	67.8	0.06	1.08	45.5
		SP+LG	0.16	4.05	23.1	0.68	24.14	16.7	0.04	1.11	53.4	0.16	2.59	31.8
		ALIKED+LG	0.37	7.57	9.2	1.43	43.02	13.6	0.13	3.27	31.4	1.82	34.59	12.7
		MASt3R	0.15	3.72	25	1.23	51.00	10.9	0.65	13.91	18.9	0.13	1.71	32.4
Canny		RoMa	1.67	38.57	20.4	1.20	30.58	32.1	0.03	0.84	54.9	0.14	1.89	31.2
		SP+LG	1.18	35.10	13.1	0.64	16.54	25.8	0.03	0.82	58.3	0.27	3.92	28.5
		ALIKED+LG	0.78	24.42	13.5	1.17	36.28	21.7	0.04	1.09	53.8	2.07	26.97	10.6
		MASt3R	0.09	2.33	25.8	0.16	3.62	18.1	0.08	1.66	38.3	0.05	0.97	48.2

Table 7. Local feature matching ablation on the Remove 360 [29, 30] dataset, using the top-20 reference images retrieved with EigenPlaces [5], and the E5+1 pose solver. We reporting median position (MPE) and orientation (MOE) errors (smaller is better) and recall (rec.) at 5 cm, 5° pose error (higher is better).

method	retrieval	day	night
original images	orig.	82.4 / 93.9 / 99.2	70.7 / 89.0 / 98.4
easy-anon	orig.	82.8 / 93.8 / 99.2	72.8 / 87.4 / 98.4
- single	obf.	82.5 / 93.7 / 99.0	71.7 / 89.0 / 98.4
easy-anon	orig.	82.4 / 93.8 / 99.0	73.3 / 88.0 / 98.4
- inpaint	obf.	81.7 / 93.8 / 99.0	71.2 / 89.5 / 98.4
SAM2	orig.	52.1 / 69.8 / 90.5	30.9 / 56.0 / 85.9
- fine masks	obf.	40.0 / 53.8 / 72.7	18.3 / 33.0 / 59.2
SAM2	orig.	59.2 / 74.5 / 91.4	36.6 / 63.4 / 89.0
- fine borders	obf.	46.4 / 58.9 / 73.8	25.7 / 39.3 / 61.3
SAM1	orig.	63.6 / 80.5 / 94.9	42.4 / 73.8 / 96.9
- fine masks	obf.	47.1 / 61.4 / 79.9	26.2 / 51.8 / 81.7
SAM1	orig.	71.7 / 83.1 / 96.1	53.9 / 80.1 / 97.4
- fine borders	obf.	51.9 / 62.9 / 77.3	34.0 / 52.9 / 74.9
Mask2Former	orig.	28.2 / 44.8 / 74.5	8.9 / 15.2 / 58.1
- semantic	obf.	13.7 / 24.8 / 46.0	3.7 / 5.2 / 19.4
Mask2Former	orig.	16.4 / 29.5 / 63.6	6.3 / 12.6 / 49.2
- random	obf.	6.8 / 12.0 / 32.5	2.1 / 4.2 / 13.6
Mask2Former	orig.	25.4 / 39.7 / 68.3	7.3 / 16.8 / 49.2
- borders	obf.	5.6 / 10.0 / 24.9	1.0 / 3.1 / 10.5
blur - 41 px	orig.	65.7 / 84.8 / 98.3	33.5 / 62.8 / 97.9
	obf.	63.3 / 79.6 / 92.2	23.6 / 44.5 / 75.4
blur - 81 px	orig.	19.3 / 44.5 / 89.4	3.1 / 22.0 / 79.6
	obf.	8.3 / 19.4 / 56.1	0.5 / 1.0 / 9.4
pixelization - 10x	orig.	50.5 / 73.3 / 96.2	24.1 / 49.2 / 86.9
	obf.	21.5 / 37.9 / 64.1	2.1 / 5.8 / 16.2
pixelization - 20x	orig.	0.7 / 7.0 / 53.3	0.0 / 0.0 / 10.5
	obf.	0.1 / 1.1 / 13.5	0.0 / 0.5 / 1.0
Canny	orig.	70.8 / 84.8 / 95.4	48.7 / 71.7 / 94.8
	obf.	52.2 / 65.0 / 78.2	24.6 / 44.0 / 58.6
DiffusionEdge	orig.	20.1 / 37.0 / 68.6	9.4 / 24.1 / 67.5
	obf.	8.7 / 16.7 / 34.0	2.1 / 6.8 / 20.9

Table 8. Localization results on Aachen Day-Night v1.1 [48, 49, 63] using the top-20 reference images retrieved with EigenPlaces [5], RoMa [16] feature matching, and pose estimation with the E5+1 solver. The retrieval is done either on the original images (orig.) or on the obfuscated images (obf.). We report the percentage of queries localized within error thresholds of (0.25 m, 2°) / (0.5 m, 5°) / (5 m, 10°).

matcher	day	night
original images		
RoMa	77.9 / 90.5 / 98.2	64.9 / **88.5** / 98.4
RoMa v2	**83.0** / **94.9** / **99.5**	**71.2** / 88.0 / **99.0**
Tiny RoMa	77.3 / 88.6 / 97.1	56.0 / 74.9 / 97.4
XFeat	55.7 / 75.6 / 94.3	18.8 / 42.9 / 87.4
Disk + LG	80.0 / 92.6 / 98.8	68.1 / 85.9 / 97.9
SAM1 - fine borders		
RoMa	**71.7** / **83.1** / **96.1**	**53.9** / **80.1** / **97.4**
RoMa v2	61.5 / 78.2 / 94.1	42.4 / 68.6 / **97.4**
Tiny RoMa	38.6 / 55.3 / 84.7	21.5 / 41.4 / 86.4
XFeat	24.6 / 41.4 / 76.8	12.0 / 26.7 / 74.3
Disk + LG	57.2 / 72.9 / 91.5	37.2 / 66.5 / 93.2

Table 9. Comparing results of pose estimation with E5+1 and different feature matching methods. The evaluation is done on Aachen Day-Night v1.1 [48, 49, 63] using top-20 retrieved reference images with EigenPlaces [5]. Reporting localization recalls (higher is better) at the pose error thresholds of (0.25 m, 2°) / (0.5 m, 5°) / (5 m, 10°).

	obfuscation	runtime
E5+1	original images	24:29
	SAM1 - fine borders	1:08:51
LT	original images	1:36:00
	SAM1 - fine borders	3:41:57

Table 10. Runtimes for the E5+1 pipeline on the Aachen Day-Night v1.1 [48, 49, 63] dataset using top-20 retrieved reference images with EigenPlaces [5]. Experiments were performed on a machine with NVIDIA A40 GPU and AMD EPYC 7543 CPU.

obfuscation	ms per image
pixelization - 20x	19
pixelization - 10x	20
Canny	32
blur - 81 px	56
blur - 41 px	133
easy-anon - single	255
easy-anon - inpaint	675
DiffusionEdge	676
SAM2 - coarse borders	999
SAM2 - coarse masks	1014
SAM1 - coarse borders	1120
SAM1 - coarse masks	1152
SAM1 - fine borders	2848
SAM1 - fine masks	2897
SAM1 - fine borders (nt)	3311
SAM1 - fine masks (nt)	3355
SAM2 - fine borders	7318
SAM2 - fine masks	7359
SAM2 - fine borders (nt)	7796
SAM2 - fine masks (nt)	7862

Table 11. Runtime comparison of different obfuscation methods averaged on a random subset of 500 images from the Aachen Day-Night v1.1 [48, 49, 63] dataset. All measurements were performed on a machine with NVIDIA A40 GPU and AMD EPYC 7543 CPU. SAM with filtered text is denoted as (nt) - no text.

type		value / unit
original images	orig.	380 kB / image
	PNG (q=75)	1058 kB / image
	JPG (q=80)	392 kB / image
SAM1 - fine masks	PNG (q=75)	366 kB / image
	JPG (q=80)	416 kB / image
SAM1 - fine borders	PNG (q=75)	257 kB / image
	JPG (q=80)	414 kB / image
	PBM + zstd	77 kB / image
	bilevel PNG	65 kB / image
ALIKED	128-D	260 B / feature
SuperPoint	256-D	516 B / feature
EigenPlaces	2048-D	4096 B / image

Table 12. Data size for different image and feature types. The data for images are averages over the Aachen Day-Night v1.1 query set at the original image size. The first row marked with "orig." are the original dataset images, where we do not have control over the used format and encoding. The "(q=N)" values specify the "-quality" parameter used during conversion with ImageMagick [25]. The values for features are theoretical data sizes based on the dimensionality of their descriptors (128-D for ALIKED and 256-D for SuperPoint) and the used data type (16-bit float). Note that the local features also contain an additional 4 bytes for keypoint coordinates.

References

[1] Luca Savant Aira, Diego Valsesia, Andrea Bordone Molini, Giulia Fracastoro, Enrico Magli, and Andrea Mirabile. Deep 3D World Models for Multi-Image Super-Resolution Beyond Optical Flow. *IEEE Access*, 2024. 2

[2] Hanadi Al-Mekhlafi and Shiguang Liu. Single Image Super-Resolution: A Comprehensive Review and Recent Insight. *Frontiers of Computer Science*, 18(1):181702, 2024. 2

[3] Anonymous. Vulnerability of privacy-preserving visual localization against diffusion-based attacks. In *Submitted to The Fourteenth International Conference on Learning Representations*, 2025. under review: https://openreview.net/forum?id=NmWf0gLufZ. 2, 3

[4] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking Visual Geo-Localization for Large-Scale Applications. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[5] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11080–11090, 2023. 6, 7, 8, 9, 10

[6] Ludovico Cavedon, Luca Foschini, and Giovanni Vigna. Getting the Face Behind the Squares: Reconstructing Pixelized Video Streams. In *5th USENIX Workshop on Offensive Technologies (WOOT 11)*, 2011. 2

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

[8] Kunal Chelani, Assia Benbihi, Fredrik Kahl, Torsten Sattler, and Zuzana Kukelova. Obfuscation Based Privacy Preserving Representations are Recoverable Using Neighborhood Information. In *2025 International Conference on 3D Vision (3DV)*, pages 189–199. IEEE, 2025. 2

[9] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating More Pixels in Image Super-Resolution Transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22367–22377, 2023. 2

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 337–33712, 2017. 5, 8

[11] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N. Sinha. Learning to Detect Scene Landmarks for Camera Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3, 5, 8, 9

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image Super-Resolution Using Deep Convolutional Networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 2

[13] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *CVPR*, pages 4829–4837, 2016. 2

[14] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss library. 2024. 5

[15] Mihai Dusmanu, Johannes L. Schonberger, Sudipta N. Sinha, and Marc Pollefeys. Privacy-preserving image features via adversarial affine subspace embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14267–14277, 2021. 2

[16] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 5, 6, 7, 8, 10

[17] Johan Edstedt, David Nordström, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. RoMa v2: Harder Better Faster Denser Feature Matching. *arXiv preprint arXiv:2511.15706*, 2025. 8

[18] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schönberger, and Marc Pollefeys. Privacy Preserving Structure-from-Motion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 333–350. Springer, 2020. 2, 7

[19] Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L. Schonberger, and Marc Pollefeys. Privacy preserving localization and mapping from uncalibrated cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1809–1819, 2021.

[20] Marcel Geppert, Viktor Larsson, Johannes L. Schönberger, and Marc Pollefeys. Privacy preserving partial localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17337–17347, 2022. 2, 7

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[22] Steven Hill, Zhimin Zhou, Lawrence Saul, and Hovav Shacham. On the (In)effectiveness of Mosaicing and Blurring as Tools for Document Redaction. *Proceedings on Privacy Enhancing Technologies*, 2016. 2

[23] Chih-Chung Hsu, Chia-Ming Lee, and Yi-Shiuan Chou. DRCT: Saving Image Super-Resolution Away from Information Bottleneck. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6133–6142, 2024. 2

[24] Mu Hu, Wei Yin, China. Xiaoyan Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2024. 2

[25] ImageMagick Studio LLC. Imagemagick. https://imagemagick.org. 10

[26] Alex Kendall, Matthew Koichi Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF

Camera Relocalization. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015. 3, 5, 6, 8

[27] Junho Kim, Changwoon Choi, Hojun Jang, and Young Min Kim. LDL: Line Distance Functions for Panoramic Localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17882–17892, 2023. 2

[28] Junho Kim, Jiwon Jeong, and Young Min Kim. Fully Geometric Panoramic Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20827–20837, 2024. 2

[29] Simona Kocour, Assia Benbihi, and Torsten Sattler. Remove360: Benchmarking Residuals After Object Removal in 3D Gaussian Splatting. *arXiv*, abs/2508.11431, 2025. 3, 5, 7, 9

[30] Simona Kocour, Assia Benbihi, and Torsten Sattler. Remove360 Dataset, 2025. https://huggingface.co/simkoc/Remove360. 3, 5, 7, 9

[31] Chunghwan Lee, Jaihoon Kim, Chanhyuk Yun, and Je Hyeong Hong. Paired-point lifting for enhanced privacy-preserving visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17266–17275, 2023. 2, 7

[32] Dawa Chyophel Lepcha, Bhawna Goyal, Ayush Dogra, and Vishal Goyal. Image Super-Resolution: A Comprehensive Review, Recent Trends, Challenges and Applications. *Information Fusion*, 91:230–260, 2023. 2

[33] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding Image Matching in 3D with MASt3R, 2024. 5

[34] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time Scene Text Detection with Differentiable Binarization. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11474–11481, 2020. 4

[35] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 5, 8

[36] Lalit Manam and Venu Madhav Govindu. Leveraging camera triplets for efficient and accurate structure-from-motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4959–4968, 2024. 5

[37] Richard McPherson, Reza Shokri, and Vitaly Shmatikov. Defeating Image Obfuscation with Deep Learning. *arXiv preprint arXiv:1609.00408*, 2016. 2

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

[39] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 4

[40] Tony Ng, Hyo Jin Kim, Vincent T. Lee, Daniel DeTone, Tsun-Yi Yang, Tianwei Shen, Eddy Ilg, Vassileios Balntas, Krystian Mikolajczyk, and Chris Sweeney. NinjaDesc: Content-Concealing Visual Descriptors via Adversarial Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12797–12807, 2022. 2

[41] Linfei Pan, Johannes L. Schönberger, Viktor Larsson, and Marc Pollefeys. Privacy preserving localization via coordinate permutations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18174–18183, 2023. 2

[42] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. SegLoc: Learning Segmentation-Based Representations for Privacy-Preserving Visual Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15380–15391, 2023. 2, 3

[43] Maxime Pietrantoni, Gabriela Csurka, and Torsten Sattler. Gaussian Splatting Feature Fields for (Privacy-Preserving) Visual Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1082–1092, 2025. 2, 3

[44] Francesco Pittaluga and Bingbing Zhuang. LDP-Feat: Image Features with Local Differential Privacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17580–17590, 2023. 2

[45] Francesco Pittaluga, Sanjeev J. Koppal, Sing Bing Kang, and Sudipta N. Sinha. Revealing Scenes by Inverting Structure From Motion Reconstructions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[46] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive Histogram Equalization and its Variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987. 4

[47] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R. Nascimento. Xfeat: Accelerated features for lightweight image matching. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2682–2691, 2024. 8

[48] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *BMVC*, 2012. 3, 7, 10

[49] 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. Benchmarking 6DOF Urban Visual Localization in Changing Conditions. In *CVPR*, 2018. 3, 7, 10

[50] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[51] Pablo Speciale, Johannes L. Schonberger, Sing Bing Kang, Sudipta N. Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 7

[52] Pablo Speciale, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy Preserving Image Queries for

Camera Localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1486–1496, 2019. 2, 7

[53] Jimmy Tekli, Bechara Al Bouna, Gilbert Tekli, and Raphaël Couturier. A Framework for Evaluating Image Obfuscation Under Deep Learning-Assisted Privacy Attacks. *Multimedia Tools and Applications*, 82(27):42173–42205, 2023. 2

[54] Alexandru Telea. An Image Inpainting Technique Based on the Fast Marching Method. *Journal of graphics tools*, 9(1): 23–34, 2004. 1

[55] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 8

[56] Shuzhe Wang, Juho Kannala, and Daniel Barath. DGC-GNN: Leveraging Geometry and Color Cues for Visual Descriptor-Free 2D-3D Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20881–20891, 2024. 2

[57] Shuzhe Wang, Juho Kannala, and Daniel Barath. DGC-GNN: leveraging geometry and color cues for visual descriptor-free 2D-3D matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20881–20891, 2024. 2

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

[59] Yuanbo Xiangli, Ruojin Cai, Hanyu Chen, Jeffrey Byrne, and Noah Snavely. Doppelgangers++: Improved visual disambiguation with geometric 3d features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 5

[60] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. EfficientSAM: Leveraged Masked Image Pretraining for Efficient Segment Anything. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16111–16121, 2024. 7

[61] Yunfan Ye, Kai Xu, Yuhang Huang, Renjiao Yi, and Zhiping Cai. DiffusionDdge: Diffusion Probabilistic Model for Crisp Edge Detection. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6675–6683, 2024. 2

[62] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *ICCV*, 2023. 2

[63] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference Pose Generation for Visual Localization via Learned Features and View Synthesis. *arXiv*, 2005.05179, 2020. 3, 7, 10

[64] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter C. Y. Chen, and Zhengguo Li. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE Transactions on Multimedia*, 2022. 5, 8

[65] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter C. Y. Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation & Measurement*, 72:1–16, 2023. 5, 8

[66] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4

[67] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic Understanding of Scenes through the ADE20K Dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 4

[68] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In *European Conference on Computer Vision*, pages 407–425. Springer, 2022. 2