Supplementary Material
LDL: Line Distance Functions for Panoramic Localization

Junho Kim[1], Changwoon Choi[1], Hojun Jang[1], and Young Min Kim[1, 2]

[1] Dept. of Electrical and Computer Engineering, Seoul National University

[2] Interdisciplinary Program in Artificial Intelligence and INMC, Seoul National University

82magnolia@snu.ac.kr, changwoon.choi00@gmail.com, {j12040208, youngmin.kim}@snu.ac.kr

A. Details on LDL

Principal Direction Computation We explain the details of principal direction computation. Recall that the principal directions in 2D and 3D are defined as the top k_{2D} and k_{3D} most common line directions. The 2D principal directions are extracted from vanishing points. When parallel lines are projected on an image, they appear to converge at a point, which is referred to as a vanishing point. To locate vanishing points, we extrapolate detected line segments and find their intersections. Since we are using panoramic images, we use spherical projection of lines and vanishing points. Specifically, we create a uniform spherical grid and count the number of intersection points in each grid cell, which we referred to as 'voting' in the main paper. We select the top k_{2D} grid locations with the most votes as the 2D principal directions. For 3D principal directions, we similarly aggregate votes for 3D line directions and extract the top k_{3D} votes. Note that we fix the filtering parameters for all our experiments and LDL achieves competitive results.

Line Filtering Prior to localization, recall from Section 3.1 that LDL filters short lines. Specifically, given the point cloud with the bounding box size of $b_x \times b_y \times b_z$, we filter out 3D line segments shorter than $\lambda(b_x + b_y + b_z)/3$, where $\lambda = 0.1$ in all our experiments. The 2D line segments are then filtered to match the filtering rate of 3D line segments. Note the threshold parameter λ does not play a critical role in performance. Figure A.1 shows the median localization error measured in Room 1 from OmniScenes [5]. The errors are nearly constant with respect to varying λ.

Spherical Quadrilateral for Computing Line Distance Functions We illustrate the spherical quadrilateral used for computing distance functions from Section 3. As shown in Figure A.2, given a line segment l on a sphere with a start point s and an end point e, the spherical quadrilateral $\mathcal{Q}(s, e)$ is formed by connecting $\{s, e, \pm(s \times e)/\|s \times e\|\}$. The spherical quadrilateral is used in Equation 2 to compute

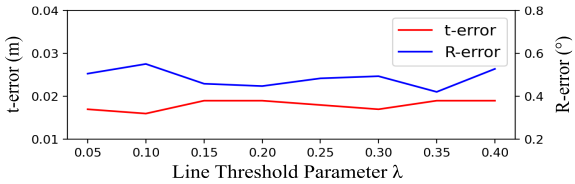
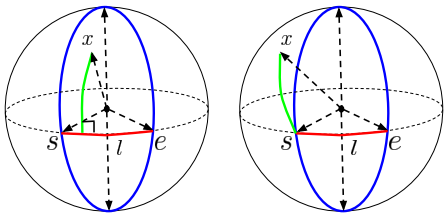


Figure A.1. Localization error against line threshold parameter λ.

the distance $D(x, l)$ from a point x to a line segment l on a sphere. Here, $D(x, l)$ is computed differently depending on whether x lies on $\mathcal{Q}(s, e)$. The 2D and 3D line distance functions (Equation 1, 3) are further built upon this definition of $D(x, l)$.

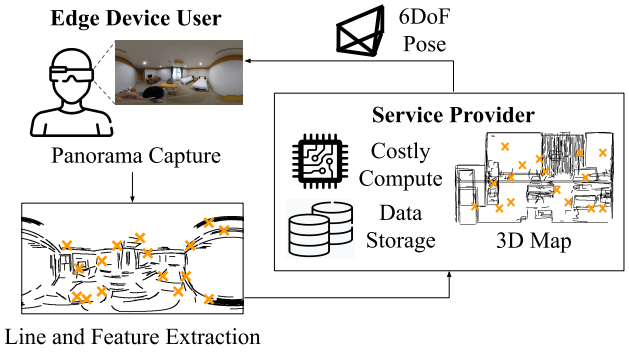
Hyperparameter Setup Here we report the hyperparameter setup of LDL. As explained in Section 3, from $N_t \times N_r$ poses we select K candidate poses by comparing the distance functions with the robust lost function in Equation 5. Recall that we use the candidate rotation estimation step in Section 3.2 to choose N_r rotations. For the N_t translations, we follow the design choice of prior works [5, 6, 12, 13] and employ uniform grid partitions for Stanford2D-3D-S [3] and centroids of octrees as in Rodenberg *et al.* [18] for OmniScenes [12]. We set K=20, N_t=800 for OmniScenes [12] and K=20, N_t=1700 for Stanford 2D-3D-S [3]. We use an increased number of translations for Stanford 2D-3D-S to cope with large scenes such as auditoriums and hallways. Nevertheless, note that LDL can quickly search promising candidate poses: even in Stanford 2D-3D-S candidate pose search finishes within 0.02 seconds.

Potential for Privacy Preservation As explained in Section 4.2, while the primary goal of LDL is to offer fast and robust localization, our approach can also be extended to offer low cost protection against various privacy breaches in client-server localization. To cope with edge devices having limited computing power, modern location-based services employ a client-server localization setup [4, 8] where

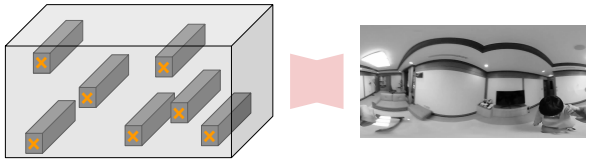


(a) $x \in \mathcal{Q}(s, e)$ (b) $x \notin \mathcal{Q}(s, e)$

Figure A.2. Given a line segment l (red), the distance (green) from point x to the l is defined depending on whether x lies on the spherical quadrilateral (blue) $\mathcal{Q}(s, e)$.



(a) LDL for Client-Server Localization



(b) Overview of Feature Inversion Attack

Figure A.3. Client-server localization setup using LDL. (a) The edge device user captures the raw 2D data and shares the lines and local features near lines with the service provider. The service provider provides the 6DoF pose using the shared information along with the 3D map. (b) While the service provider can attempt feature inversion attacks by training neural networks that learn image reconstructions from local feature inputs, this cannot fully recover the sensitive visual details for LDL as only a fraction of information is shared.

the visual data of the edge device is shared with the service provider [8, 17]. Based on the shared information, the service provider performs the actual localization pipeline and returns the estimated 6DoF pose to the edge device user.

We adapt LDL to the client-server localization scenario while offering privacy protection by having the edge device user to only share lines and local features near lines during localization. Specifically, as shown in Figure A.3, we modify the pose refinement phase of LDL to operate using local features near lines, instead of all the visible local features used for the original refinement explained in Section 3.4. Here we only consider line segments whose lengths are over a designated threshold as explained in Section 3.1 and directions are parallel to one of the 2D principal directions. Such a modification results in privacy protection against feature inversion attacks [8, 16, 17], which take local feature vectors as input and outputs an image reconstruction. Note that LDL naturally offers privacy protection during pose selection as it only uses line segments for this phase and thus does not necessitate the clients to share their entire view with the service provider. We further demonstrate the potential of LDL for privacy protection through experiments shown in Section B.5.

Area	t-error (m)		R-error (°)		Accuracy	
	LDL	LDL$^{\text{LS}}$	LDL	LDL$^{\text{LS}}$	LDL	LDL$^{\text{LS}}$
Area 1	**0.02**	**0.02**	**0.54**	0.60	**0.86**	0.75
Area 2	**0.02**	0.05	**0.66**	0.79	**0.77**	0.57
Area 3	**0.02**	0.03	**0.54**	0.73	**0.89**	0.69
Area 4	**0.02**	**0.02**	**0.48**	0.57	**0.88**	0.72
Area 5	**0.02**	0.03	**0.54**	0.61	**0.81**	0.59
Area 6	**0.02**	**0.02**	**0.50**	0.58	**0.83**	0.66
Total	**0.02**	0.03	**0.53**	0.64	**0.83**	0.66

Table B.1. Ablation study of uniformly sampling query points on the unit sphere. LDL is compared against a variant using query points sampled along 2D line segment locations (LDL$^{\text{LS}}$) in the Stanford 2D-3D-S dataset [3].

B. Additional Experimental Results

B.1. Additional Ablation Study

Choice of Query Point Locations We report the impact of choosing uniformly sampled query points for evaluating distance functions. Recall that we rank $N_t \times N_r$ poses with the robust loss function in Equation 5, where the query points Q are uniformly sampled from a unit sphere. We compare LDL against a variant that uses query points sampled along the 2D line segment locations. Namely, this variant only considers regions with line segments, in contrast to LDL that equally considers regions lacking lines.

We make quantitative evaluations between LDL and the variant using the Stanford 2D-3D-S [3] dataset. For fair comparison, we use identical hyperparameters as the original implementation of LDL. As shown in Table B.1, uniform sampling employed in LDL leads to large amounts of performance improvement. By fairly using all regions on the sphere, LDL effectively utilizes the spatial context from the line distance functions and performs effective localization.

Choice of Loss Function We validate the robust loss function in Equation 5 by comparing LDL against variants using other loss functions: L1, L2, Huber, and Median loss. Here we report results from the Wedding Hall scene in OmniScenes, as this scene contains drastic scene changes with

Method	t-error (m)	R-error (°)	Acc.
w/ L1 Loss	0.08	1.38	0.55
w/ L2 Loss	0.17	1.48	0.34
w/ Huber Loss	0.11	1.39	0.50
w/ Median Loss	0.08	**1.22**	0.55
Ours	**0.07**	**1.22**	**0.68**

Table B.2. Ablation study on the choice of loss functions evaluated in OmniScenes [12].

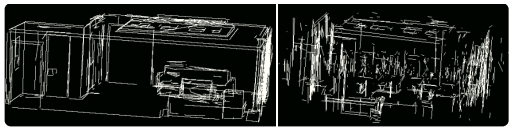


Figure B.4. 3D Lines from 3D Scanning (Left) and SfM (Right).

Method	t-error (m)	R-error (°)	Acc.
SfM	**0.03**	0.80	0.85
3D Scan	**0.03**	**0.71**	**0.98**

Table B.3. Evaluation results of LDL on noisier line maps obtained using structure from motion and Line3D++ [11].

Figure B.5. Top-down view of offices in Stanford 2D-3D-S [3].

large amounts of outliers. As shown in Table B.2, the inlier counting proposed in Equation 5 attenuates outliers in the Extreme split and exhibits optimal performance, demonstrating the effectiveness of the robust loss function.

B.2. Evaluation in Noisier Maps

In the main paper, we extract 3D lines from point clouds obtained using Matterport 3D scanners [1]. Here we run LDL on noisier line maps created using structure-from-motion (SfM) and Line3D++ [11]. As shown in Figure B.4, the maps are more noisier than those from 3D scans. Table B.3 shows the localization results from Room 3, 5 in Omniscenes under different types of line maps (note the new pipeline did not produce reliable maps in other scenes). Even though LDL was run with the exact same hyperparameters as in the main paper, it shows only a small amount of performance drop, which indicates that it can robustly handle noisier SfM-based line maps which are generated without 3D scanners.

B.3. Additional Evaluation in Large-Scale Maps

In the main paper we demonstrated that LDL can perform competitively against the structure-based method in large scenes by testing multiple room localization in OmniScenes [23]. To further show the scalability of LDL, we evaluate on 20 office rooms from Stanford 2D-3D-S [3], and localize each image against the jointly composed 3D map. The 20 office rooms contain similar structures, as shown in Figure B.5. Even in such conditions, LDL shows similar performance against the structure-based method as shown in Table B.4. While scalability has not been the main goal of this paper, LDL shows the potential to be deployed in large-scale localization settings containing visual ambiguities.

Method	t-error (m)	R-error (°)	Acc.
LDL	**0.02**	**0.54**	**0.90**
Structure-Based	0.03	0.58	0.89

Table B.4. Evaluation on Large Scale Scenes

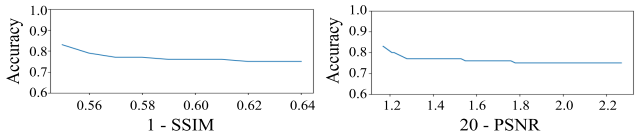


Figure B.6. Privacy-utility curve drawn from various values of line-based filtering thresholds in the Stanford 2D-3D-S dataset. While the reconstruction quality of feature inversion attacks largely degrade as we filter out more feature points, the localization accuracy remains relatively constant.

B.4. Full Localization Evaluation Results at Various Accuracy Thresholds

We share the full localization results for the OmniScenes [12] and Stanford 2D-3D-S [3] datasets in Table B.5, B.6. Here we additionally show the localization accuracy at various accuracy thresholds. Our method can perform competitively against all the tested baselines across various thresholds, while performing light-weight pose search with line distance functions.

B.5. Privacy Preservation Results

We share the detailed privacy evaluation results on the Stanford 2D-3D-S [21] dataset. Table B.7 shows the localization accuracy along with the feature inversion attack results. The image error metrics (20 - PSNR, 1-SSIM, MAE) of the feature inversion attacks measured against the original panorama consistently increase for all tested scenarios, indicating that our line-based feature filtering can successfully hide visual details. This notion is further verified through the additional qualitative samples in Figure B.7 where the sensitive visual data such as tabletop clutter are removed after filtering. Nevertheless, note that the filtering process only incurs a small drop in localization performance. We finally evaluate how LDL balances privacy (feature inversion protection) and utility (localization accuracy) while using line-based feature filtering. In Figure B.6

Accuracy (0.05 m, 5°)	PC	CPO	SB	LT	CD	LDL
Robot	0.66	0.88	0.86	0.85	0.27	**0.92**
Hand	0.77	0.77	0.73	0.72	0.22	**0.82**
Change Robot	0.39	0.58	0.72	0.72	0.21	**0.78**
Change Hand	0.45	0.58	0.68	0.70	0.22	**0.72**
Extreme	0.38	0.57	0.63	0.62	0.20	**0.71**

(a) Accuracy at translation and rotation threshold 0.05 m, 5°

Accuracy (0.05 m, 10°)	PC	CPO	SB	LT	CD	LDL
Robot	0.66	0.88	0.86	0.85	0.27	**0.92**
Hand	0.77	0.77	0.73	0.72	0.22	**0.82**
Change Robot	0.39	0.58	0.72	0.72	0.21	**0.78**
Change Hand	0.45	0.58	0.68	0.70	0.22	**0.72**
Extreme	0.38	0.57	0.63	0.62	0.20	**0.71**

(b) Accuracy at translation and rotation threshold 0.05 m, 10°

Accuracy (0.1 m, 5°)	PC	CPO	SB	LT	CD	LDL
Robot	0.69	0.89	**0.99**	**0.99**	0.31	**0.98**
Hand	0.81	0.80	0.95	0.95	0.29	**0.97**
Change Robot	0.41	0.59	0.93	0.94	0.30	**0.95**
Change Hand	0.47	0.60	**0.92**	0.90	0.30	**0.92**
Extreme	0.41	0.59	0.89	0.88	0.29	**0.92**

(c) Accuracy at translation and rotation threshold 0.1 m, 5°

Accuracy (0.1 m, 10°)	PC	CPO	SB	LT	CD	LDL
Robot	0.69	0.89	**0.99**	**0.99**	0.32	**0.98**
Hand	0.81	0.80	0.95	0.95	0.29	**0.97**
Change Robot	0.41	0.59	0.93	0.94	0.30	**0.95**
Change Hand	0.47	0.60	**0.92**	0.90	0.30	**0.92**
Extreme	0.41	0.59	0.89	0.88	0.29	**0.92**

(d) Accuracy at translation and rotation threshold 0.1 m, 10°

Accuracy (0.2 m, 5°)	PC	CPO	SB	LT	CD	LDL
Robot	0.70	0.89	**1.00**	**1.00**	0.34	**0.99**
Hand	0.81	0.81	0.98	0.98	0.32	**0.99**
Change Robot	0.41	0.59	0.98	**0.99**	0.33	**0.98**
Change Hand	0.48	0.60	**0.97**	**0.97**	0.34	**0.97**
Extreme	0.42	0.60	0.96	0.96	0.34	**0.98**

(e) Accuracy at translation and rotation threshold 0.2 m, 5°

Accuracy (0.2 m, 10°)	PC	CPO	SB	LT	CD	LDL
Robot	0.70	0.89	**1.00**	**1.00**	0.34	**0.99**
Hand	0.81	0.81	0.98	0.98	0.33	**0.99**
Change Robot	0.41	0.59	0.98	**0.99**	0.33	**0.98**
Change Hand	0.49	0.60	**0.97**	**0.97**	0.34	**0.97**
Extreme	0.42	0.60	0.96	0.96	0.34	**0.98**

(f) Accuracy at translation and rotation threshold 0.2 m, 10°

Table B.5. Localization accuracy at various thresholds in the OmniScenes [12] dataset.

we plot the image error metrics of feature inversion attacks against the original image along with the localization accuracy using various line-based filtering threshold values. While the discrepancy values increase largely, the localization accuracy remains relatively constant. Thus the modified version of LDL can balance between privacy protection and accurate localization, suggesting its future potential as a robust privacy-preserving localization algorithm.

Nevertheless, the current modification cannot fully hide keypoints from large structures such as walls and ceilings. While these regions typically do not contain sensitive visual information, some users may want their entire views to

Accuracy (0.05 m, 5°)	PC	CPO	SB	LT	CD	LDL
Area 1	0.66	**0.89**	0.83	0.83	0.46	0.83
Area 2	0.42	**0.81**	0.63	0.63	0.30	0.69
Area 3	0.53	0.76	0.81	0.82	0.34	**0.86**
Area 4	0.48	0.83	0.87	**0.88**	0.43	0.85
Area 5	0.44	0.73	0.68	0.69	0.34	**0.74**
Area 6	0.68	**0.90**	0.80	0.82	0.45	0.81

(a) Accuracy at translation and rotation threshold 0.05 m, 5°

Accuracy (0.05 m, 10°)	PC	CPO	SB	LT	CD	LDL
Area 1	0.66	**0.90**	0.83	0.83	0.46	0.83
Area 2	0.42	**0.81**	0.63	0.63	0.30	0.69
Area 3	0.53	0.76	0.81	0.82	0.34	**0.86**
Area 4	0.48	0.83	0.87	**0.88**	0.43	0.85
Area 5	0.44	0.73	0.68	0.69	0.34	**0.74**
Area 6	0.68	**0.90**	0.80	0.82	0.45	0.81

(b) Accuracy at translation and rotation threshold 0.05 m, 10°

Accuracy (0.1 m, 5°)	PC	CPO	SB	LT	CD	LDL
Area 1	0.66	**0.90**	0.89	**0.90**	0.50	0.86
Area 2	0.45	**0.81**	0.76	0.74	0.35	0.77
Area 3	0.57	0.78	**0.92**	0.88	0.36	**0.89**
Area 4	0.49	0.83	**0.91**	**0.91**	0.46	0.88
Area 5	0.44	0.74	0.80	0.79	0.36	**0.81**
Area 6	0.69	**0.90**	0.88	0.87	0.47	0.83

(c) Accuracy at translation and rotation threshold 0.1 m, 5°

Accuracy (0.1 m, 10°)	PC	CPO	SB	LT	CD	LDL
Area 1	0.66	**0.90**	0.89	**0.90**	0.50	0.86
Area 2	0.45	**0.81**	0.76	0.74	0.35	0.77
Area 3	0.57	0.78	**0.92**	0.88	0.36	**0.89**
Area 4	0.49	0.83	**0.91**	**0.91**	0.46	0.88
Area 5	0.44	0.74	0.80	0.79	0.36	**0.81**
Area 6	0.69	**0.90**	0.88	0.87	0.47	0.83

(d) Accuracy at translation and rotation threshold 0.1 m, 10°

Accuracy (0.2 m, 5°)	PC	CPO	SB	LT	CD	LDL
Area 1	0.67	**0.90**	0.89	**0.90**	0.50	0.86
Area 2	0.47	**0.81**	0.80	**0.81**	0.37	0.78
Area 3	0.59	0.81	**0.96**	0.93	0.41	**0.95**
Area 4	0.50	0.83	**0.94**	**0.93**	0.47	0.89
Area 5	0.47	0.78	**0.84**	**0.84**	0.39	**0.84**
Area 6	0.69	**0.90**	0.88	0.88	0.48	0.84

(e) Accuracy at translation and rotation threshold 0.2 m, 5°

Accuracy (0.2 m, 10°)	PC	CPO	SB	LT	CD	LDL
Area 1	0.67	**0.90**	0.89	**0.90**	0.50	0.86
Area 2	0.47	**0.81**	0.80	**0.81**	0.37	0.78
Area 3	0.59	0.81	**0.96**	0.93	0.41	**0.95**
Area 4	0.50	0.83	**0.94**	**0.93**	0.47	0.89
Area 5	0.47	0.78	**0.84**	**0.84**	0.39	**0.84**
Area 6	0.69	**0.90**	0.88	0.88	0.48	0.84

(f) Accuracy at translation and rotation threshold 0.2 m, 10°

Table B.6. Localization accuracy at various thresholds in the Stanford 2D-3D-S [12] dataset.

be hidden from service providers. Developing a more secure line-based localization algorithm that could alleviate a wider range of concerns is left as future work.

C. Baseline Details

In this section, we describe the details for implementing the baselines compared against LDL. We implement PIC-

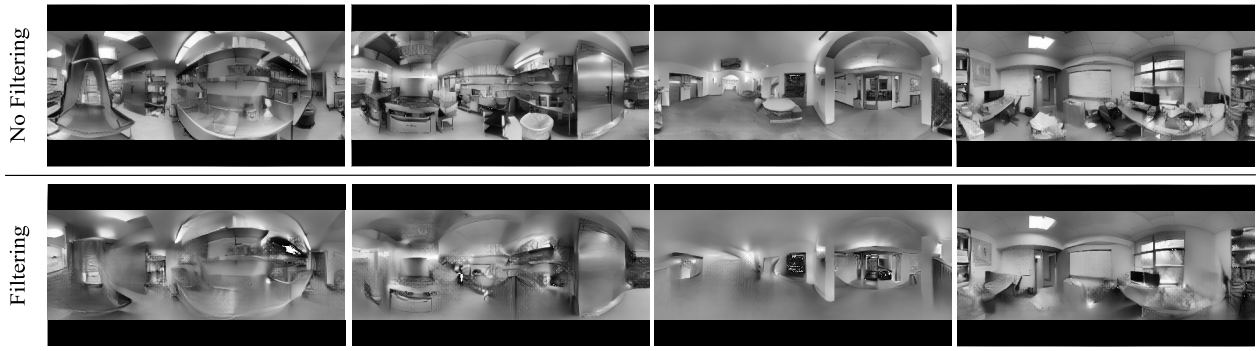


Figure B.7. Deletion of objects in feature inversion attacks after line-based filtering.

Method	t-error (m)	R-error (°)	Acc.
No Filtering	**0.02**	**0.53**	**0.83**
Filtering	0.03	0.64	0.77

(a) Localization Performance Evaluation in Stanford 2D-3D-S [3]

Method	20-PSNR	1-SSIM	MAE
No Filtering	1.1717	0.5505	0.1598
Filtering	**1.7577**	**0.6027**	**0.1773**

(b) Reconstruction Quality of Feature Inversion Attacks

Table B.7. Privacy-preservation evaluation of modified LDL using line-based feature filtering evaluated in Stanford 2D-3D-S dataset [3]. The simple filtering incurs only a small drop in localization accuracy while largely increasing the image error metrics.

COLO [12] and CPO [13] from the publicly available codebase. Below we retain our description on the Structure-based, Chamfer-based, and Line Transformer-based approaches.

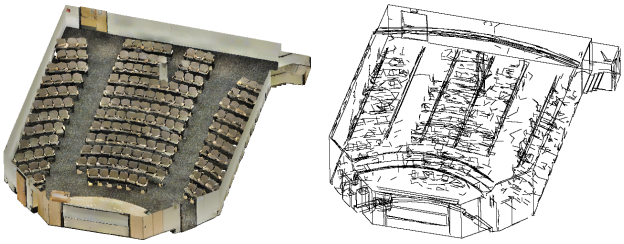
Structure-Based Approach As explained in Section 4, structured-based approach first finds promising candidate poses using robust image retrieval and then refines poses using PnP-RANSAC from feature matches. For image retrieval we use NetVLAD [2], which is a widely used image retrieval method that outputs a global feature vector for each image. To deploy NetVLAD in our setup, we first render $N_t \times N_r$ synthetic views from the point cloud. Here we use $N_t = 100$ candidate translations and $N_r = 216$ candidate rotations uniformly sampled from $SO(3)$. Then, we extract the global features for each synthetic view and the query image, and choose the top $K = 20$ synthetic views whose feature vectors are closest to the query image. As the final step, we perform feature matching [19] from each selected synthetic view against the query image, and choose the final view with the most matches. To ensure fair comparison, we undistort the selected view and the query panorama into cubemaps and separately perform feature matching for each pair of faces. The matches are then aggregated to perform refinement via PnP-RANSAC [10].

Chamfer Distance-Based Approach Inspired from Micusik et al. [15], Chamfer distance-based approach first selects poses that best align 3D lines against lines in the query image, where the Chamfer distance is used to evaluate the potential matchings. The selected poses are then refined with PnP-RANSAC, similar to the structure-based approach. To elaborate, we find the top $K = 20$ poses from an initial pool of $N_t \times N_r$ poses, where the poses are ranked by measuring the Chamfer distance between the projected line segments in 3D and those in the query image. We set N_t and N_r identical to LDL and use the principal directions for deducing a set of candidate rotations. As the final step, we render views at the selected K poses and perform feature matching against the query image for refinement via PnP-RANSAC.

Line Transformer-Based Approach Based on Yoon et al. [22], Line Transformer-based approach finds candidate poses attaining the most line matches with the query image, and refines poses using PnP-RANSAC. For establishing line matches, we first render $N_t \times N_r$ synthetic views from the point cloud where we set $N_t = 100$ and $N_r = 216$. Then, the top $K_1 = 100$ poses are selected whose NetVLAD [2] features are closest to the query image. This intermediate step is necessary as the line transformer features are computationally expensive and thus could not be naively evaluated for all $N_t \times N_r$ views. For each synthetic view from the selected poses, we extract line Transformer embeddings and establish matchings with the query image. Similar to the structure-based baseline, we convert panoramas to cubemaps during the line matching process. Finally, we select the top $K_2 = 20$ poses that have the most line matches, and refine them via PnP-RANSAC.

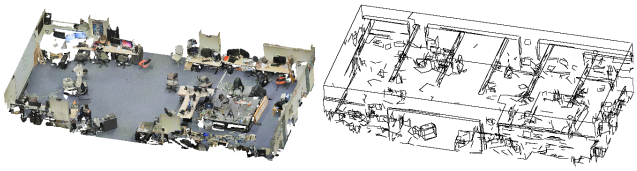
D. Details on Experimental Setup

In this section, we provide additional details for experiments presented in Section 4 and Section B.



Original 3D Point Cloud Detected 3D Lines

(a) Auditorium 1 from Stanford 2D-3D-S Dataset



Original 3D Point Cloud Detected 3D Lines

(b) Office 31 from Stanford 2D-3D-S Dataset

Figure D.8. Visualization of the 3D line segments used for LDL. While the line segment extraction algorithm from Xiaohu et al. [20] can reliably extract the wireframe-like structure from the original 3D scan, the line segments are still quite noisy. Note that we have cropped the ceilings of the original point cloud for better visualization.

Illumination Robustness Evaluation To evaluate the robustness of LDL against illumination shifts, we apply synthetic color variations to images in Room 3 from OmniScenes [12]. We consider three synthetic color variations, where qualitative examples are shown in Figure 4: average intensity, gamma, and white balance change. For average intensity change we lower each pixel intensity by 25%. For gamma change, we set the image gamma to 0.2. For white balance change, we apply the following transformation matrix to the raw RGB color values: $\begin{pmatrix} 0.7 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.8 \end{pmatrix}$.

Candidate Pose Search Evaluation We compare LDL against NetVLAD [2] for candidate pose search using the Extreme split from OmniScenes. The recall curves in Figure 5 are obtained by measuring the localization performance of both methods prior to pose refinement. As mentioned in Section 4.2, we use the identical set of translations with $N_t = 50$ for both methods and associate a large number of candidate rotations $N_r = 216$ for NetVLAD to ensure fair comparison. Such measures are taken for rotations since LDL estimates rotations using combinatorial matchings of principal directions, which makes the number of candidate rotations to vary for each query image. We empirically find that less than 30 candidate rotations remain after discarding infeasible rotations, and thus setting $N_r = 216$ for NetVLAD would provide enough evidence to achieve competitive performance against LDL.

Feature Inversion Network for Privacy Evaluation To evaluate the privacy protection of LDL against feature inversion attacks, we train a fully-convolutional neural network $F_\Theta(\cdot)$ that takes a sparse feature map $D \in \mathbb{R}^{H \times W \times C}$ as input and produces image reconstructions. The feature map stores local feature descriptors $\mathbf{f} \in \mathbb{R}^C$ at keypoint locations $(i_{\text{kpt}}, j_{\text{kpt}})$, namely $D(i_{\text{kpt}}, j_{\text{kpt}}) = \mathbf{f}$, and zero values for other regions. For the inversion network, we use a similar U-Net architecture as in Ng et al. [16] where the only difference is in the input channel dimension that we set as 256 instead of 128 to match the SuperPoint [9] descriptor dimensions. Then for training, we use the entire Matterport3D [7] dataset where we use the first 90% of the 9581 panorama images for training and the rest for validation. We follow the training procedure of Ng et al. [16] and use the perceptual loss and mean absolute error (MAE) loss, where we employ Adam [14] with a learning rate of $1e{-}4$ for optimization. In our experiments, we use the trained network to reconstruct panoramas from the local feature descriptors, where we shared the reconstruction results along with the image error metrics in Section 4 and Section B. To elaborate, during evaluation we first extract local features for each query image in the Stanford 2D-3D-S dataset [3] and run feature inversion, where the results are then compared against the original panorama image.

3D Line Maps for Localization In Figure D.8, we show visualizations of 3D lines used as input to LDL. Despite the reliabilty of the 3D line extraction algorithm of Xiaohu et al. [20], the lines are still quite noisy. To cope with the noisy detections, LDL employs a length-based filtering scheme to only keep long, salient lines and resorts to matching the *distribution* of lines using line distance functions instead of trying to establish direct one-to-one matchings as in previous works [15, 22].

References

[1] Matterport 3d: How long does it take to scan a property? https://support.matterport.com/hc/en-us/articles/229136307-How-long-does-it-take-to-scan-a-property-. Accessed: 2020-02-18.

[2] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[3] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[4] Christian Cachin, Idit Keidar, and Alexander Shraer. Trusting the cloud. *SIGACT News*, 40(2):81–86, jun 2009.

[5] Dylan Campbell, Lars Petersson, Laurent Kneip, and Hongdong Li. Globally-optimal inlier set maximisation for camera pose and correspondence estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page preprint, June 2018.

[6] Dylan Campbell, Lars Petersson, Laurent Kneip, Hongdong Li, and Stephen Gould. The alignment of the spheres: Globally-optimal spherical mixture alignment for camera pose estimation. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page to appear, Long Beach, USA, June 2019. IEEE.

[7] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[8] Deeksha Dangwal, Vincent T. Lee, Hyo Jin Kim, Tianwei Shen, Meghan Cowan, Rajvi Shah, Caroline Trippel, Brandon Reagen, Timothy Sherwood, Vasileios Balntas, Armin Alaghi, and Eddy Ilg. Analysis and mitigations of reverse engineering attacks on local feature descriptors, 2021.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR Deep Learning for Visual SLAM Workshop*, 2018.

[10] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[11] Manuel Hofer, Michael Maurer, and Horst Bischof. Efficient 3d scene abstraction using line segments. *Computer Vision and Image Understanding*, 157:167–178, 2017. Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans.

[12] Junho Kim, Changwoon Choi, Hojun Jang, and Young Min Kim. Piccolo: Point cloud-centric omnidirectional localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3313–3323, October 2021.

[13] Junho Kim, Hojun Jang, Changwoon Choi, and Young Min Kim. Cpo: Change robust panorama to point cloud localization. ECCV, 2022.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[15] Branislav Micusik and Horst Wildenauer. Descriptor free visual indoor localization with line segments. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3165–3173, 2015.

[16] 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, June 2022.

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

[18] O. B. P. M. Rodenberg, E. Verbree, and S. Zlatanova. Indoor A* Pathfinding Through an Octree Representation of a Point Cloud. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV21:249–255, Oct. 2016.

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

[20] Lu Xiaohu, Liu Yahui, and Li Kai. Fast 3d line segment detection from unorganized point cloud. *arXiv preprint arXiv:1901.02532*, 2019.

[21] W. Yang, Y. Qian, J. Kämäräinen, F. Cricri, and L. Fan. Object detection in equirectangular panorama. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2190–2195, Aug 2018.

[22] Sungho Yoon and Ayoung Kim. Line as a visual sentence: Context-aware line descriptor for visual localization. *IEEE Robotics and Automation Letters*, 6(4):8726–8733, 2021.