Indoor Visual Localization using Point and Line Correspondences in dense colored point cloud

Yuya Matsumoto
NEC Corporation
yuya-matsumoto@nec.com

Gaku Nakano
NEC Corporation
g-nakano@nec.com

Kazumine Ogura
NEC Corporation
k-oguraay@nec.com

Abstract

We propose a novel pipeline called Loc-PL that uses both points and lines for indoor visual localization in dense colored point cloud. Loc-PL utilizes the spatially complementary relationship between points and lines to address challenging indoor issues. There are two successive camera pose estimation modules. The first improves robustness against repetitive patterns by considering the geometric consistency of points and lines. The second utilizes points and lines to refine poses by Perspective-m-Point-n-Line (PmPnL) and circumvents unstable localization due to locally concentrated matches caused by less-textured environments. The modules use different schemes to obtain line correspondences; the first finds line matches using RANSAC, which is effective for image pairs with large viewpoint gaps, and the second utilizes rendered images from dense point cloud to get them by feature line matching. In addition, we develop a simple but effective module for evaluating the correctness of camera poses using matched point distances across two images. The experimental results on a large dataset, InLoc, show that Loc-PL achieves the state-of-the-art in four out of six scores.

1. Introduction

Visual localization is a technique for estimating the 6-DoF camera pose of a query image in a pre-constructed 3D space. It has been widely used for various applications such as augmented reality and robot navigation [7–9, 25].

It is well known that localization in indoor environments is a more challenging task than in outdoor ones due to the presence of texture-less scenes and repetitive patterns. Most localization approaches are based on the detection of point correspondences and the Perspective-n-Point (PnP) algorithm. However, keypoints are hard to detect in texture-less areas and are sometimes locally concentrated, which makes localization unstable. Also, repetitive patterns cause false correspondences. Therefore, point-based approaches often suffer from these indoor properties.

In comparison, indoor environments have linear structures in general. Since lines are detected even in texture-less scenes, points and lines have spatially complementary relationships.

This complementarity of points and lines motivated us to develop a novel pipeline, Loc-PL, for indoor visual localization, which is shown in Figure 1. Our pipeline consists of two camera pose estimation modules using both points and lines. The two respective modules play different roles in localization and use a different strategy to obtain line correspondences. The first module takes the consistency of points and lines into account to make localization robust to repetitive patterns. It finds line correspondences with a RANSAC-based approach that is effective for image pairs with large viewpoint gaps (Sec. 4.2). The second module utilizes points and lines to circumvent locally concentrated correspondences. This module obtains line correspondences using rendered images from dense point cloud, and Perspective-m-Point-n-Line (PmPnL) enables stable refinement of camera poses by using points and lines. Both modules utilize the complementarity of points and lines and are effectively incorporated into well-developed coarse-to-fine localization pipelines. Also, we developed a simple module that evaluates the correctness of estimated camera poses to improve localization accuracy (Sec. 4.3).

In our experiments on a large-scale indoor dataset, InLoc, our pipeline achieved the state-of-the-art (SOTA) in four out of six scores. Moreover, quantitative and qualitative ablation studies report that all the three modules improve localization accuracy, and the two different schemes for finding line correspondences are very effective.

This paper is organized as follows. Section 2 introduces the conventional approaches for indoor visual localization. Section 3 reviews related works using line features. Section 4 first describes the baseline method and then gives details on the proposed algorithms. Section 5 reports the experimental results. Finally, Section 6 summarizes our conclusion.

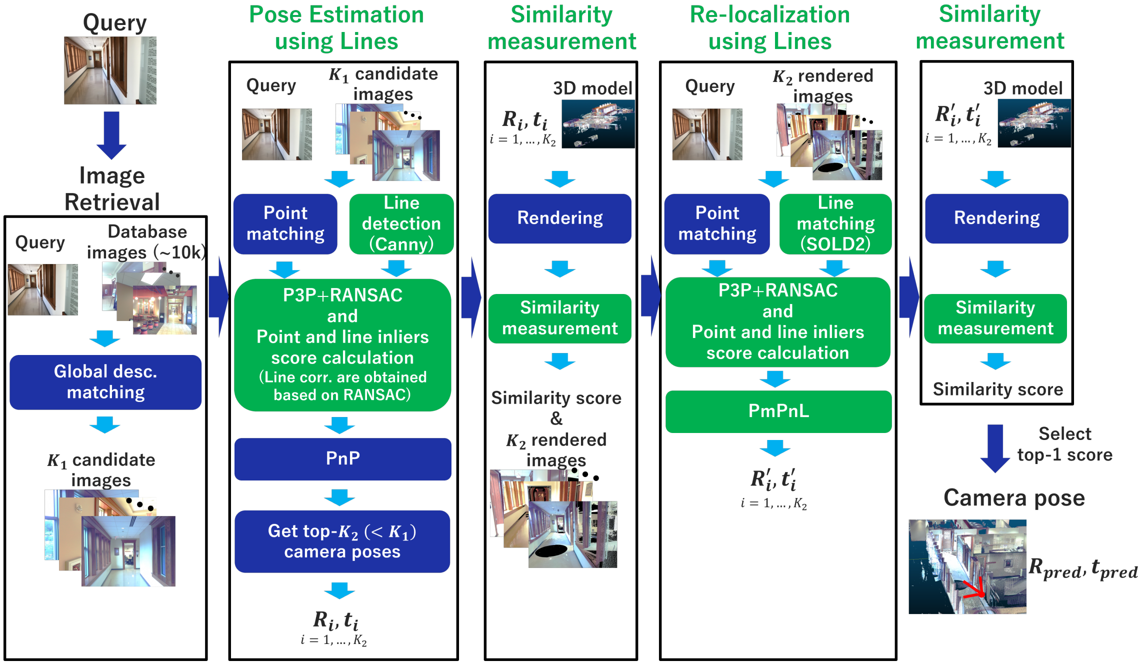


Figure 1. Whole pipeline of proposed coarse-to-fine approach using both point and line correspondences. Novel modules are colored in green. Pose estimation step utilizes points and lines to make localization robust to repetitive patterns (Sec. 4.2). Distance-based similarity measurement provides more accurate scores than DenseRootSIFT-based approach (Sec. 4.3). Spatially distributed point and line correspondences stabilize refinement of camera poses by Perspective-m-Point-n-Line (PmPnL) in re-localization step (Sec. 4.4).

2. Indoor Visual Localization

A lot of approaches have been developed for visual localization. Some methods are based on direct regression [6, 17, 18]; however, it has been pointed out that these methods have poor generalization performance due to their scene-specific approach [31]. Currently, the state-of-the-art methods for large-scale datasets are mostly coarse-to-fine models [29, 35]. This approach begins by efficiently retrieving a small set of images similar to a query from a large image database. Then, point feature matching is performed between the query and retrieved images one by one to obtain 2D-2D point correspondences. Since the 2D pixels of the database images are associated with corresponding 3D points by SfM and LiDAR, we obtain 2D-3D correspondences by using them. Finally, camera poses are given by solving the PnP problem.

Indoor visual localization has been considered a difficult task compared with outdoor localization because of the presence of less-textured environments, repetitive patterns, dynamic objects, *etc*. Deep-learning techniques have been introduced to improve each step of coarse-to-fine model such as image retrieval [4, 14, 27] and feature point matching [10, 30, 34]. Dense matching [35] is used for obtaining point correspondences in texture-less areas. To remove feature points on dynamic objects, semantic segmentation [11] is used to mask them. The above methods are based on keypoint matching; therefore, they are essentially not robust against less-textured scenes and repetitive patterns.

Several studies have attempted to improve robustness to indoor scenes by matching line segments between images, including chamfer loss minimization [24], vanishing points of parallel lines [43], and 2D-3D line matching [44]. PtLine [12] utilizes both points and lines for visual localization on a large-scale indoor dataset. This method performs camera pose estimation twice; a query pose is first estimated, and then estimation is repeated by using the once-estimated pose to refine it. Lines are only used in the second estimation in this method. In our pipeline, lines are utilized in both steps not only to refine poses but also to make localization robust to repetitive patterns. Also, PtLine obtains line correspondences on the basis of an epipolar constraint between the initially estimated pose and database images' poses. In comparison, we get line matches with a RANSAC-based approach in the first estimation and with visual line descriptors in the second one, which is explained in detail in Sec. 4.2 and Sec. 4.4.

Another challenging issue with indoor visual localization is the sparsity of image viewpoints in a database. If database images are constructed with a fixed 3D scanner like InLoc, the sparsity becomes high and degrades the accuracy of image retrieval and camera pose estimation.

To fill the sparsity gap, several studies project dense 3D point cloud onto new viewpoints and add the rendered images to the database [40, 46]. In another study, keypoints across multiple images are utilized for correcting camera poses [16].

3. Related Work

This section briefly reviews recent studies on line features and camera pose estimation using line segments.

Line detection The mainstream of hand-crafted methods is to use low-level features or image gradients for line detection (e.g., LSD [39], EDLines [2], Cannyline [23]). Since low-level features are susceptible to illumination changes and camera angles, maintaining high repeatability is one of the main issues. Another limitation is that the global context of an image are not used due to local gradient analysis. After wireframe parsing [15] was introduced, several deep-learning methods have been proposed to extract structural line segments. The first method to use wireframe parsing [15] predicts two maps, a junction map and a line heat map, and extracts line segments from them on the basis of a heuristic approach. Several improvements have been proposed, such as an end-to-end method [49], an attraction field map prediction [41, 42], and a trainable Hough transform [22].

Line description One of the most popular line descriptors is LBD [47], which is a hand-crafted feature based on the image gradient around line segments. LJL [21] is another hand-crafted feature focusing on the intersection of adjacent lines. Learning-based descriptors have also been developed. DLD [19] is based on distance learning using triplet loss. LLD [38] is a computationally efficient descriptor designed for Visual SLAM. Several methods for simultaneously detecting and describing lines have been proposed, such as SOLD2 [26], ELSD [45], and L2D2 [1]. Although extensive efforts have been made, the matching accuracy is not as good as that of feature points.

Perspective-n-Point (Line) problem The PnP problem uses n point correspondences to estimate a camera pose. Line correspondences are also used to estimate camera poses in the Perspective-n-Line (PnL) algorithm. In recent years, PnL solvers have attracted attention as a method for preserving privacy in SfM [13], Visual SLAM [32], and camera localization [33]. Moreover, both point and line correspondences can be combined to estimate camera poses [3, 37, 48].

4. Visual Localization using Lines

In this section, we start by defining a baseline method using only point correspondences. There are two camera pose estimation modules, namely, pose estimation and re-localization. Then, we introduce new schemes utilizing line correspondences into the respective modules in our proposal. The entire proposed pipeline is shown in Fig. 1. The usage of lines in the first module makes localization robust to repetitive patterns. In the second module, spatially distributed correspondences of points and lines stabilize camera pose estimation by PmPnL.

4.1. Baseline using only points

The baseline method is based on a pipeline of the coarse-to-fine model, which uses only point correspondences. Its architecture consists of four modules:

- **Image retrieval** This step searches top-K_1 images that are similar to a query image from a large-scale image database. For computational efficiency, each image is converted to a global feature vector in general. The similarity between the query and database images is calculated by the Euclidean distance.
- **Pose estimation** Feature point matching is carried out between the query image and the K_1 images to obtain 2D-2D point correspondences. Each pixel of the database images is associated with a corresponding 3D point; therefore, the query image has 2D-3D point correspondences for K_1 images. Camera poses are estimated by solving the PnP problem incorporated with P3P+RANSAC for outlier rejection. Among the K_1 camera poses, top-K_2 ones are selected by scoring the number of inliers in the 2D-3D point correspondences.
- **Similarity measurement** The points of a pre-constructed 3D map are projected onto the K_2 cameras to generate rendered images. Then, the similarity between the query and each of them is measured. In this paper, we compare Modified Pose Verification [16] and a proposed simple approach based on 2D-2D point matching, which is described in Sec. 4.3.
- **Re-localization** The K_2 camera poses are re-localized by conducting the above two steps (pose estimation and similarity measurement) again. Then, each camera pose and its similarity score are updated if the new score is better than the previous one. Finally, the pipeline outputs a camera pose with the maximum similarity as the estimated query pose.

4.2. Pose estimation using points and lines

Point-based methods often suffer due to repetitive patterns in indoor visual localization, which cause false point correspondences. Figure 2 shows an example of failure due to these patterns. It is essentially hard for point-based approaches to circumvent this problem.

To address this, we utilize line correspondences in the first camera pose estimation to make localization robust to repetitive patterns. Our approach takes the consistency of

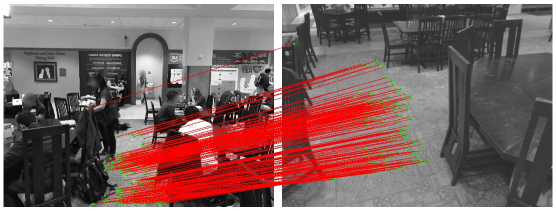


Figure 2. Example of false point correspondences due to repetitive patterns.

points and lines into account for the selection of similar database images to query during RANSAC loop.

First, we detect 2D line segments L_q and L_d from a query and K_1 database images, respectively, along with feature point matching. Here, L is a set of detected line segments. We determine 3D line segments corresponding to L_d using the 3D points in the database. They are projected onto the query image with the camera pose $\boldsymbol{R}_t, \boldsymbol{t}_t$ estimated by P3P+RANSAC and the intrinsic parameter $\boldsymbol{K}$. Then, we obtain line segments L'_d, *i.e*. the projection of L_d to the query. The distance between two line segments $l_q \in L_q$ and $l'_d \in L'_d$ is given by

$$d = \frac{|Au^s_d + Bv^s_d + C| + |Au^e_d + Bv^e_d + C|}{\sqrt{A^2 + B^2}}, \quad (1)$$

where (A, B, C) denotes the coefficients of an infinite line containing l_q, and (u_d, v_d) represents the 2D coordinates of the two endpoints of l'_d indexed by the superscripts s and e. Let $\boldsymbol{v}_q$ and $\boldsymbol{v}'_d$ be the direction vectors of l_q and l'_d, respectively. The angle between l_q and l'_d can be written as

$$\theta = \cos^{-1}(\boldsymbol{v}_q^T \boldsymbol{v}'_d). \quad (2)$$

We calculate the distance d and the angle θ for all possible combinations of L_q and L_d. Then, we select inlier line segments that satisfy two conditions: 1) (d, θ) are less than a threshold (d_{th}, θ_{th}), and 2) l_q and l_d are mutually nearest neighbors in the query and database images. This procedure is conducted along with inlier points search in a RANSAC loop.

After obtaining N_p point inliers and N_l line inliers, we determine a RANSAC score S by

$$S = (N_p + 1) \times (N_l + 1). \quad (3)$$

The adding one is done merely to prevent a zero value when $N_p = 0$ or $N_l = 0$. The global line structure in 3D space generally varies depending on a scene. Therefore, even if a repetitive pattern yields a large N_p in a local area of a wrong scene, we can expect Eq. (3) to be small because the scene would give a small N_l. Finally, we refine the camera pose by PnP using point correspondences.

This exhaustive search based on RANSAC to obtain line correspondences is more successful than feature line matching in this initial estimation step :line matching is still a very difficult task, and existing models based on visual line descriptors cannot find correct and sufficient line correspondences from images with large viewpoint changes. This is discussed again in the ablation study in Sec. 5.4.

4.3. Similarity measurement

DenseRootSIFT [5] has been used in recent methods for measuring the similarity between a query and rendered images in visual localization [16, 35]. In [16], the usage of DenseRootSIFT is customized and referred to as Modified Pose Verification (MPV). However, SIFT features are sensitive to blank pixels in a rendered image or brightness differences between several 3D scans for the same scene. Recent learning-based features are more robust to such noise in images because they utilize keypoint locations and the context of the descriptors for point matching. This motivated us to develop a simple method for calculating similarity scores using SuperPoint [10] and SuperGlue [30]. We first obtain 2D-2D point correspondences between a query and a rendered image and then calculate the similarity score by counting the number of point correspondences that satisfy the distance threshold d_p. Despite the straightforward approach, it is a reasonable measure because the query and the rendered image are very similar if the camera pose of the query image is correctly estimated. We refer to this simple procedure as PointMatching.

4.4. Re-localization using points and lines

Several studies [16, 46] report that camera re-localization, *i.e*., estimating camera poses repeatedly, is effective for improving localization accuracy. Conventional methods use only point correspondences to perform it. However, few keypoints are detected in less-textured areas and are often locally concentrated in an image, which is insufficient for refining camera poses.

We use line correspondences as well as points in the re-localization module to refine camera poses. Since lines also appear in texture-less scenes, points and lines are detected complementarily in an image. Spatially distributed correspondences, in general, make camera pose optimization stable and correct. We utilize both types of correspondences for optimization by using Perspective-m-Point-n-Line (PmPnL).

In this module, we use visual line descriptors to obtain line correspondences, unlike the pose estimation module. In the re-localization step, the existing line matching models work well since a query and rendered images should have close viewpoints if camera poses are correctly localized. After obtaining initial matches of the points and lines, we conduct a similar approach to the pose estimation described in Sec. 4.2 to remove outlier points and lines. During a P3P+RANSAC loop, line inliers are selected on the basis of Eqs. (1) and (2), and the RANSAC score is calculated by Eq. (3). Note that we do not perform the exhaustive search for line correspondences, unlike Sec. 4.2 because the initial line matches are given by the line descriptors.

After the RANSAC scheme, we have the inliers of 2D-3D point and line correspondences. Then, we project 3D points onto the query image, and the reprojection error of an i-th point pair can be given by

$$e_i = |\boldsymbol{p}_i - \boldsymbol{p}'_i|, \tag{4}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{p}'_i$ are a 2D point detected in the query image and a projected point of the corresponding 3D point, respectively. Also, the reprojection error of a j-th line pair for the two endpoints s and e can be written as

$$\begin{aligned} d_j^s &= \frac{|Au^s + Bv^s + C|}{\sqrt{A^2 + B^2}}, \\ d_j^e &= \frac{|Au^e + Bv^e + C|}{\sqrt{A^2 + B^2}}, \end{aligned} \tag{5}$$

where (A, B, C) denotes the coefficients of an infinite line containing the 2D line of the query, and (u^s, v^s) and (u^e, v^e) represent the 2D coordinates of the two endpoints of a line projected from the 3D line segment. Then, the optimal camera pose $\boldsymbol{R}^*, \boldsymbol{t}^*$ can be given by minimizing the total cost function below:

$$\boldsymbol{R}^*, \boldsymbol{t}^* = \underset{\boldsymbol{R} \in \mathrm{SO}(3), \boldsymbol{t} \in \mathrm{R}^3}{\arg\min} \frac{1}{M} \sum_{i=1}^{M} |e_i|^2 + \frac{1}{N} \sum_{j=1}^{N} \left(|d_j^s|^2 + |d_j^e|^2 \right) \tag{6}$$

where M and N are the number of the pairs of points and lines, respectively. We refer Eq. (6) to the PmPnL problem. The initial guess of $\boldsymbol{R}$ and $\boldsymbol{t}$ is given by P3P+RANSAC, and the Levenberg-Marquardt method is used for the optimization. Since we can obtain spatially distributed points and lines even in a less-textured scene, camera pose estimation becomes stable compared with the estimation using only points. In Sec. 5.4, we will compare the accuracy of PmPnL with that of PnP.

5. Experiments

5.1. Dataset

We used the InLoc dataset [35] for evaluating the proposed method in the experiment. InLoc is a large-scale indoor dataset for visual localization that covers a wide area of $25,287\text{m}^2$. It consists of five floors (DUC1, DUC2, CSE3, CSE4, and CSE5) and provides 10K images in total. Each pixel of the images is associated with a corresponding 3D point measured by a 3D LiDAR sensor. There are 329 query images in total for DUC1 and DUC2 captured by iPhone 7.

To evaluate performance, localization accuracy is measured by the percentage of estimated poses whose positional errors are less than thresholds of 0.25 / 0.50 / 1.00 meters and whose angular errors are within $10°$. All evaluations were conducted on an online benchmark server for visual localization tasks[1]. The re-localization step in our pipeline needs rendered images from dense point clouds. To the best of our knowledge, InLoc is the only dataset with such dense point cloud, and many papers use only this dataset to evaluate of localization [12, 35, 36, 46].

5.2. Implementation

In all steps, the learning-based models were not fine-tuned. Publicly available models were used.

Image retrieval We used NetVLAD [4] with a VGG16 backbone pre-trained on the Pitts30K dataset for global image descriptors. Images were resized so that the longer side of either the height or width was 640 pixels. The number of candidate images was set to $K_1 = 100$

Pose estimation We detected feature points by SuperPoint [10] and SuperGlue [30] pre-trained on the MegaDepth dataset. For line segment detection, we used fastLineDetector [20] on OpenCV, which is a variant of the Canny edge detector. Fragmented lines shorter than 100 pixels in length were discarded. Images were resized so that the longer side of either the height or width was 1200 pixels. RANSAC was configured to have a 10-pixel threshold for point correspondences. In addition, the thresholds for line correspondences were set to $d_{th} = 20$ pixels and $\theta_{th} = 2°$. The number of new candidates selected from K_1 images was set to $K_2 = 10$.

Similarity measurement The resolution of rendered images was 640 pixels for the longer side of either the height or width to fill pixels as much as possible. The distance threshold for PointMatching was $d_p = 5$ pixels.

Re-localization The query image was resized to the same resolution as the rendered images. SuperPoint and SuperGlue were used again for point feature matching. Since the image size is smaller than the pose estimation step, we set smaller thresholds for RANSAC iterations: 5 pixels for point correspondences, $d_{th} = 10$ pixels, and $\theta_{th} = 1°$ for line correspondences. Also, we used SuperGlue pre-trained on ScanNet since its recommended size of input images is suitable for smaller input images. For obtaining line correspondences, SOLD2 [26] was used.

5.3. Comparison with the state-of-the-art methods

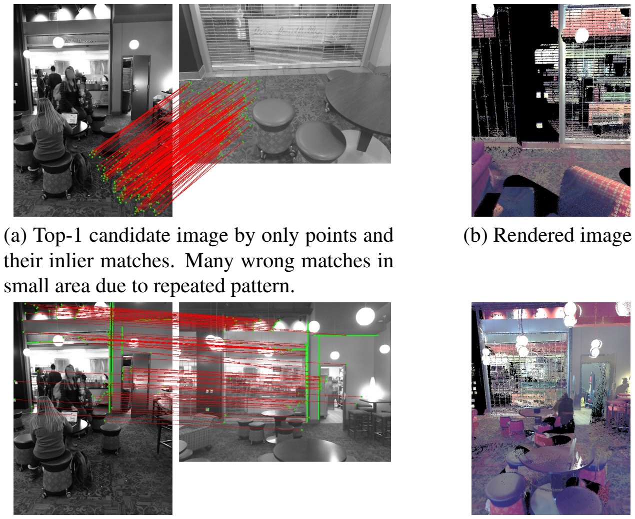
We quantitatively compared our proposal with the existing SOTA methods: InLoc [35], PtLine [12] and its baseline using only points, RenderNet [46], and PCLoc [16]. PtLine is a method that uses points and lines and was compared

[1] https://www.visuallocalization.net/

Method	Features		DUC1			DUC2		
	Points	Lines	0.25m	0.50m	1.00m	0.25m	0.50m	1.00m
InLoc [35]	NetVLAD [4]	–	40.9	58.1	70.2	35.9	54.2	69.5
PtLine Baseline [12]	SuperPoint, R2D2 [28]	–	47.0	71.2	84.8	61.1	77.9	80.2
PtLine [12]	SuperPoint, R2D2 [28]	VLSE [12]	50.5	72.7	86.9	61.8	79.4	84.0
RenderNet [46]	SuperPoint, RenderNetL	–	58.6	77.8	89.4	74.8	82.4	85.5
PCLoc (4096) [16]	SuperPoint	–	60.6	79.8	90.4	70.2	92.4	93.1
PCLoc (3000) [16]	SuperPoint	–	59.6	78.3	89.4	71.0	**93.1**	**93.9**
Baseline	SuperPoint	–	62.1	79.3	87.9	74.0	83.2	84.7
Ours (Loc-PL)	SuperPoint	Canny [20], SOLD2 [26]	**64.6**	**84.8**	**93.4**	**76.3**	86.3	88.5

Table 1. Quantitative comparison of proposed method with existing approaches on InLoc dataset. Best and second-best values are in bold and underlined, respectively. *Features* represents feature correspondences used in each method.

with its baseline using only points in its paper. PCLoc has two variations depending on the number of feature points used, namely, 3000 and 4096. Table 1 indicates that the proposed method achieved SOTA performance in four out of six scores. In particular, for the most strict criterion of < 0.25 meters, the proposed method is better than PCLoc (3000) by 5% for DUC1 and RenderNet by 1.5% for DUC2. Our proposal achieves SOTA on all DUC1 scores. Also, the improvement for our proposal from the baseline using points is larger than that of PtLine: the improvements for the average six scores of PtLine and ours are 2.18% and 3.78%, and the maximum improvements among the six scores are 3.8% and 5.5%, respectively. These results show that our pipeline effectively utilizes points and lines.


(a) Top-1 candidate image by only points and their inlier matches. Many wrong matches in small area due to repeated pattern.

(b) Rendered image

(c) Top-1 candidate image by points and lines, and their inlier matches. Points and lines are correctly matched.

(d) Rendered image

Figure 3. (a) Top-1 candidate image wrongly selected by only point correspondences due to repetitive patterns, and (b) rendered image obtained by camera pose estimated from candidate image. We can see that the pose estimation failed since view of rendered image is different from that of query. (c) Top-1 candidate image correctly selected by both point and line correspondences, and (d) rendered view obtained by camera pose estimated from image. Query and rendered view have same view, and camera pose estimation is successful.

5.4. Ablation study

5.4.1 Effectiveness of lines

We conducted ablation studies to investigate the effectiveness of line correspondences in our proposal. Table 2 reports the quantitative results of eight possible combinations of the implementations: 1) with or without line correspondences in the pose estimation, 2) MPV [16] or PointMatching in the similarity measurement, and 3) PnP or PmPnL optimization in the re-localization.

First, the use of line correspondences in the pose estimation improves all scores for < 0.5 and < 1.0 meters compared with the case of only points. Figure 3 shows a qualitative result with or without line correspondences in a scene with a repetitive pattern. Calculating inlier scores using only point correspondences results in a localized concentration of inliers at a wrong location, and camera pose estimation fails. In comparison, introducing line correspondences makes localization take linear structures into account in the RANSAC loop. This circumvents false correspondences due to repetitive patterns.

Moreover, at the re-localization step, the camera pose optimization with PmPnL using both point and line correspondences tends to increase scores for $<$ 0.25 and 0.50 meters. Since points and lines appear complementarily in images, we can obtain spatially distributed correspondences to make camera pose optimization more stable. Figure 4 visualizes a comparison of PnP with only point correspondences and PmPnL with both points and lines. As shown in Fig. 4b, most of the inlier points in the point-based re-localization are located around the windows far from the camera position. On the other hand, in Fig. 4c, there are line matches as well as points on the ceiling near the cam-

Pose estimation	Similarity measurement	Re-localization	DUC1			DUC2		
			0.25m	0.50m	1.00m	0.25m	0.50m	1.00m
Points (Baseline)	MPV	PnP	62.1	79.3	87.9	74.0	83.2	84.7
Points	MPV	PmPnL	63.6	78.3	87.9	**76.3**	85.5	87.0
Points	PointMatching	PnP	63.6	82.3	91.9	68.7	83.2	86.3
Points	PointMatching	PmPnL	**65.2**	79.8	91.9	72.5	84.7	86.3
Points + Lines	MPV	PnP	62.6	79.8	87.9	74.0	84.7	85.5
Points + Lines	MPV	PmPnL	61.1	79.3	88.4	73.3	**86.3**	87.0
Points + Lines	PointMatching	PnP	**65.2**	82.8	**93.4**	71.0	83.2	87.0
Points + Lines	PointMatching	PmPnL	64.6	**84.8**	**93.4**	**76.3**	**86.3**	**88.5**

Table 2. Ablation study on proposed method. Experiments were conducted on $2 \times 2 \times 2 = 8$ variations: 1) with or without line correspondences in pose estimation, 2) MPV [16] or PointMatching in similarity measurement, and 3) PnP or PmPnL optimization in re-localization. Best and second-best values are in bold and underlined, respectively.

Pose estimation	Re-localization	DUC1			DUC2			Average
		0.25m	0.50m	1.00m	0.25m	0.50m	1.00m	
RANSAC	Feature line matching	**64.6**	**84.8**	**93.4**	**76.3**	**86.3**	**88.5**	**82.3**
RANSAC	RANSAC	62.6	81.8	**93.4**	73.3	84.0	86.3	80.2
Feature line matching	Feature line matching	59.1	80.3	90.4	74.8	84.7	87.0	79.4
Feature line matching	RANSAC	59.1	80.3	89.4	71.8	84.7	87.0	78.7

Table 3. Ablation study on comparison between the RANSAC-based exhaustive search and feature line matching (SOLD2) to obtain line correspondences.

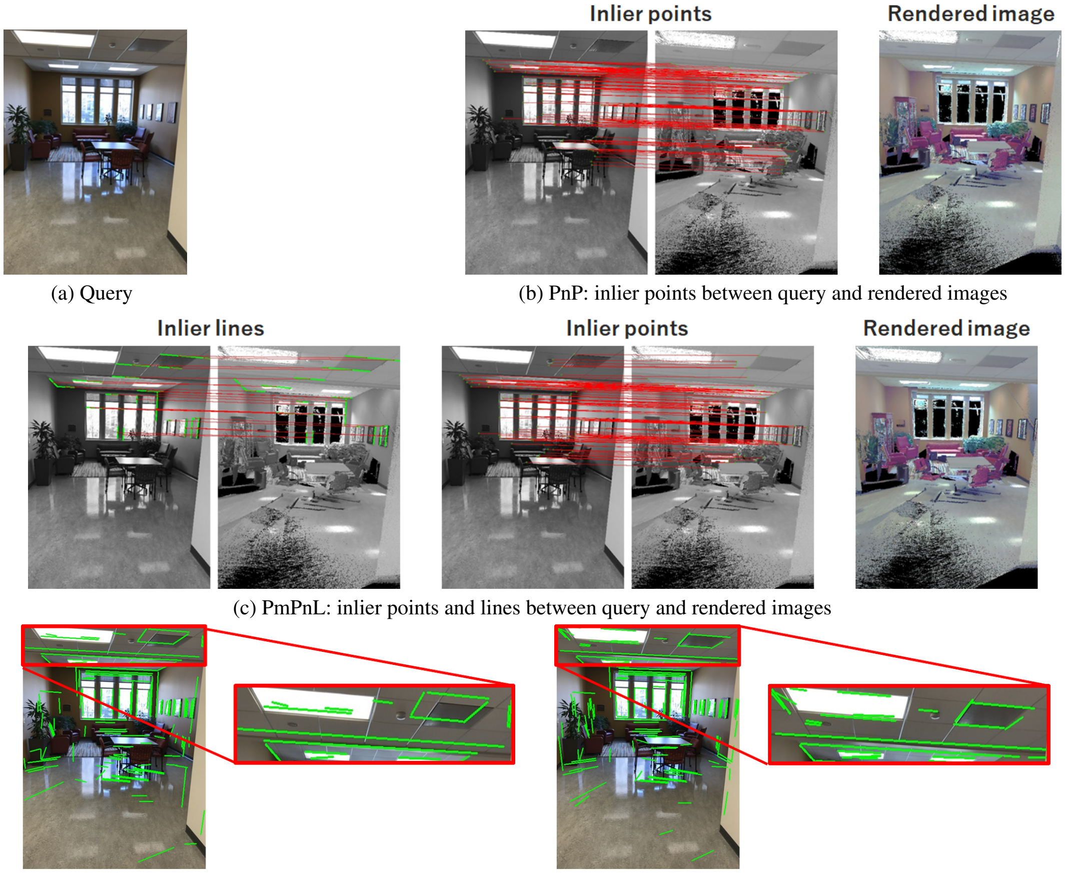
era when combining lines. Also, point correspondences are removed around dynamic objects such as the desk and chair. Figs. 4d and 4e show the edges of the rendered images overlaid on the query image by PnP and PmPnL, respectively. We can observe that the edges by PnP are misaligned near the ceiling while PmPnL utilizes both points and lines to realize correct estimation.

5.4.2 RANSAC-based method vs. feature line matching for obtaining line correspondences

In our pipeline, we obtain line correspondences by exhaustive search based on RANSAC in the first pose estimation and by a feature line matching model (SOLD2) in the second. To show that this strategy is effective, we investigated the changes to performance when using the two methods in each of the two estimation modules. Table 3 shows the localization results of the respective approaches, where the RANSAC-based method and feature matching use fastLineDetector and SOLD2, respectively, as they do in our pipeline. In addition, some examples of line inliers with the respective methods are shown in Figure 5. We see an overall decrease in accuracy when using the feature line matching in the pose estimation instead of the RANSAC-based one: query and database images often have a large viewpoint gap in the first estimation, and existing line matching models still don't have enough performance to find line matches in such a case. Also, the use of the RANSAC-based method in the re-localization step lowers the scores within 0.25m and 0.50m. This method determines line matches on the basis of the angle and distance of lines on a 2D image plane. Therefore, it cannot consider visual features and depth, which can cause false matches. Since image pairs in the second step have similar views, the existing feature line matching models work better than the RANSAC-based one.

6. Conclusion

We have presented a novel method utilizing the complementarity of points and lines for indoor visual localization. Our pipeline estimates camera poses in two successive modules using both points and lines. The first module makes localization robust to repetitive patterns, and the second one stabilizes camera pose optimization to refine poses. These estimation modules use different schemes to obtain line correspondences, namely, a RANSAC-based method and feature line matching considering the viewpoint changes of input image pairs. Also, we introduced a simple method for measuring the correctness of estimated poses. We demonstrated in experiments on a large-scale dataset, InLoc, that the combination of the three proposed algorithms is qualitatively and quantitatively superior to the existing methods.



(a) Query

(b) PnP: inlier points between query and rendered images

(c) PmPnL: inlier points and lines between query and rendered images

(d) PnP: edges in rendered image overlaid on query image

(e) PmPnL: edges in rendered image overlaid on query image

Figure 4. Inlier points or lines determined by PnP and PmPnL solvers at re-localization step.

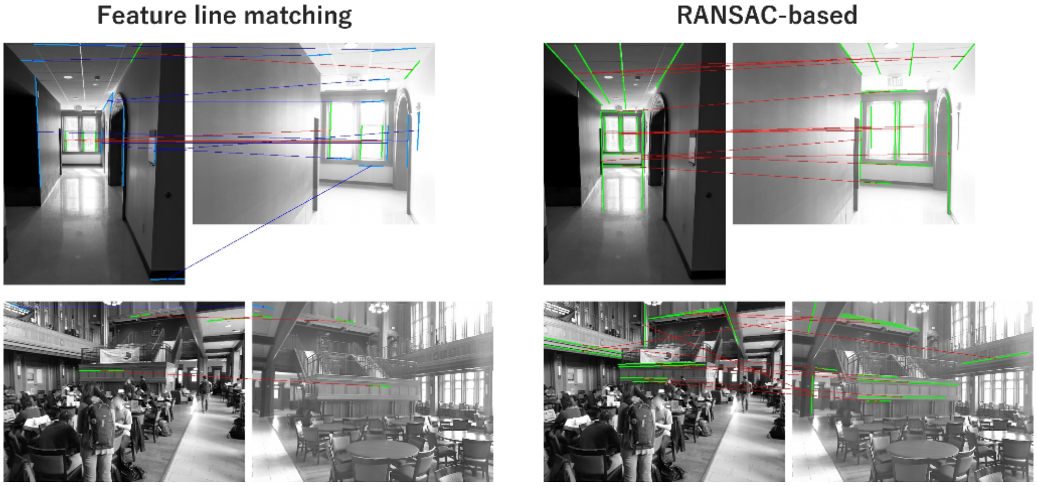


Figure 5. Two examples of comparison between the exhaustive search based RANSAC and feature line matching in the pose estimation step. Green lines represent the inliers of RANSAC, and they are connected with red lines. Light blue represents lines that are matched by feature line matching but not inliers in RANSAC, connected with blue. Existing line matching models cannot find sufficient and correct line correspondences, while RANSAC-based one work well.

References

[1] Hichem Abdellali, Robert Frohlich, Viktor Vilagos, and Zoltan Kato. L2d2: Learnable line detector and descriptor. In *2021 International Conference on 3D Vision (3DV)*, pages 442–452. IEEE, 2021. 3

[2] Cuneyt Akinlar and Cihan Topal. Edlines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633–1642, 2011. 3

[3] Adnan Ansar and Konstantinos Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003. 3

[4] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 2, 5, 6

[5] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2911–2918. IEEE, 2012. 4

[6] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2616–2625, 2018. 2

[7] Robert Castle, Georg Klein, and David W Murray. Video-rate localization in multiple maps for wearable augmented reality. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 15–22. IEEE, 2008. 1

[8] Wendy H Chun and Tobias Höllerer. Real-time hand interaction for augmented reality on mobile phones. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 307–314, 2013. 1

[9] Andrzej Debski, Wojciech Grajewski, Wojciech Zaborowski, and Wojciech Turek. Open-source localization device for indoor mobile robots. *Procedia Computer Science*, 76:139–146, 2015. 1

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2, 4, 5

[11] Martina Dubenova, Anna Zderadickova, Ondrej Kafka, Tomas Pajdla, and Michal Polic. D-inloc++: Indoor localization in dynamic environments. In *Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings*, pages 246–261. Springer, 2022. 2

[12] Shuang Gao, Jixiang Wan, Yishan Ping, Xudong Zhang, Shuzhou Dong, Yuchen Yang, Haikuan Ning, Jijunnan Li, and Yandong Guo. Pose refinement with joint optimization of visual points and lines. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2888–2894. IEEE, 2022. 2, 5, 6

[13] 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. 3

[14] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021. 2

[15] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018. 3

[16] Janghun Hyeon, Joohyung Kim, and Nakju Doh. Pose correction for highly accurate visual localization in large-scale indoor spaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15974–15983, 2021. 3, 4, 5, 6, 7

[17] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5974–5983, 2017. 2

[18] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2

[19] Manuel Lange, Fabian Schweinfurth, and Andreas Schilling. Dld: A deep learning based line descriptor for line feature matching. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5910–5915. IEEE, 2019. 3

[20] Jin Han Lee, Sehyung Lee, Guoxuan Zhang, Jongwoo Lim, Wan Kyun Chung, and Il Hong Suh. Outdoor place recognition in urban environments using straight lines. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5550–5557. IEEE, 2014. 5, 6

[21] Kai Li, Jian Yao, Xiaohu Lu, Li Li, and Zhichao Zhang. Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation. *Neurocomputing*, 184:207–220, 2016. 3

[22] Yancong Lin, Silvia L Pintea, and Jan C van Gemert. Deep hough-transform line priors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 323–340. Springer, 2020. 3

[23] Xiaohu Lu, Jian Yao, Kai Li, and Li Li. Cannylines: A parameter-free line segment detector. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 507–511. IEEE, 2015. 3

[24] Branislav Micusik and Horst Wildenauer. Descriptor free visual indoor localization with line segments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3165–3173, 2015. 2

[25] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pages 268–283. Springer, 2014. 1

[26] Rémi Pautrat, Juan-Ting Lin, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Sold2: Self-supervised occlusion-aware line description and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11368–11378, 2021. 3, 5, 6

[27] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. 2

[28] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32, 2019. 6

[29] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 2

[30] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 4, 5

[31] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3302–3312, 2019. 2

[32] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy preserving visual slam. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 102–118. Springer, 2020. 3

[33] 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*, pages 5493–5503, 2019. 3

[34] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 2

[35] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7199–7209, 2018. 2, 4, 5, 6

[36] Hajime Taira, Ignacio Rocco, Jiri Sedlar, Masatoshi Okutomi, Josef Sivic, Tomas Pajdla, Torsten Sattler, and Akihiko Torii. Is this the right place? geometric-semantic pose verification for indoor visual localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4373–4383, 2019. 5

[37] Alexander Vakhitov, Jan Funke, and Francesc Moreno-Noguer. Accurate and linear time pose estimation from points and lines. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII*, pages 583–599. Springer, 2016. 3

[38] Alexander Vakhitov and Victor Lempitsky. Learnable line segment descriptor for visual slam. *IEEE Access*, 7:39923–39934, 2019. 3

[39] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):722–732, 2008. 3

[40] Hui-Xuan Wang, Jing-Liang Peng, Shi-Yi Lu, Xin Cao, Xue-Ying Qin, and Chang-He Tu. Reloc: indoor visual localization with hierarchical sitemap and view synthesis. *Journal of Computer Science and Technology*, 36(3):494–507, 2021. 3

[41] Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, and Liangpei Zhang. Learning attraction field representation for robust line segment detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1595–1603, 2019. 3

[42] Nan Xue, Tianfu Wu, Song Bai, Fudong Wang, Gui-Song Xia, Liangpei Zhang, and Philip HS Torr. Holistically-attracted wireframe parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2788–2797, 2020. 3

[43] Huai Yu, Weikun Zhen, Wen Yang, and Sebastian Scherer. Line-based 2-d–3-d registration and camera localization in structured environments. *IEEE Transactions on Instrumentation and Measurement*, 69(11):8962–8972, 2020. 2

[44] Huai Yu, Weikun Zhen, Wen Yang, Ji Zhang, and Sebastian Scherer. Monocular camera localization in prior lidar maps with 2d-3d line correspondences. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4588–4594. IEEE, 2020. 2

[45] Haotian Zhang, Yicheng Luo, Fangbo Qin, Yijia He, and Xiao Liu. Elsd: efficient line segment detector and descriptor. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2969–2978, 2021. 3

[46] Jiahui Zhang, Shitao Tang, Kejie Qiu, Rui Huang, Chuan Fang, Le Cui, Zilong Dong, Siyu Zhu, and Ping Tan. Rendernet: Visual relocalization using virtual viewpoints in large-scale indoor environments. *arXiv preprint arXiv:2207.12579*, 2022. 3, 4, 5, 6

[47] Lilian Zhang and Reinhard Koch. An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. *Journal of visual communication and image representation*, 24(7):794–805, 2013. 3

[48] Lipu Zhou, Jiamin Ye, and Michael Kaess. A stable algebraic camera pose estimation for minimal configurations of 2d/3d point and line correspondences. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV 14*, pages 273–288. Springer, 2019. 3

[49] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 962–971, 2019. 3