3D Line Mapping Revisited

Shaohui Liu[1] Yifan Yu[1] Rémi Pautrat[1] Marc Pollefeys[1,2] Viktor Larsson[3]
[1]Department of Computer Science, ETH Zurich [2]Microsoft [3]Lund University

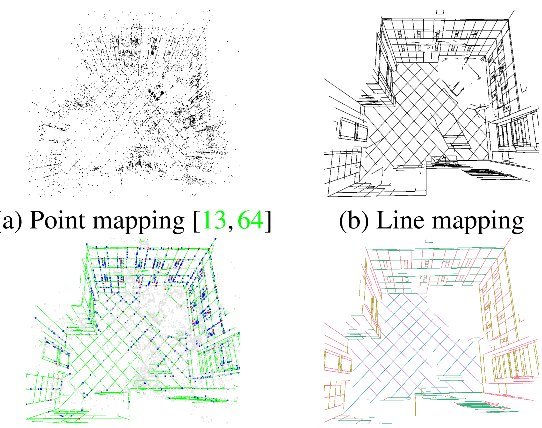
Abstract

In contrast to sparse keypoints, a handful of line segments can concisely encode the high-level scene layout, as they often delineate the main structural elements. In addition to offering strong geometric cues, they are also omnipresent in urban landscapes and indoor scenes. Despite their apparent advantages, current line-based reconstruction methods are far behind their point-based counterparts. In this paper we aim to close the gap by introducing LIMAP, a library for 3D line mapping that robustly and efficiently creates 3D line maps from multi-view imagery. This is achieved through revisiting the degeneracy problem of line triangulation, carefully crafted scoring and track building, and exploiting structural priors such as line coincidence, parallelism, and orthogonality. Our code integrates seamlessly with existing point-based Structure-from-Motion methods and can leverage their 3D points to further improve the line reconstruction. Furthermore, as a byproduct, the method is able to recover 3D association graphs between lines and points / vanishing points (VPs). In thorough experiments, we show that LIMAP significantly outperforms existing approaches for 3D line mapping. Our robust 3D line maps also open up new research directions. We show two example applications: visual localization and bundle adjustment, where integrating lines alongside points yields the best results. Code is available at https://github.com/cvg/limap.

1. Introduction

The ability to estimate 3D geometry and build sparse maps via Structure-from-Motion (SfM) has become ubiquitous in 3D computer vision. These frameworks enable important tasks such as building maps for localization [60], providing initial estimates for dense reconstruction and refinement [65], and novel view synthesis [45, 48]. Currently, the field is dominated by point-based methods in which 2D keypoints are detected, matched, and triangulated into 3D maps [20, 64]. These sparse maps offer a compact scene representation, only reconstructing the most distinctive points.

While there have been tremendous progress in point-based reconstruction methods, they still struggle in scenes



(a) Point mapping [13, 64] (b) Line mapping
(c) Line-point association (d) Line-VP association

Figure 1. In this paper, we propose a robust pipeline for mapping 3D lines (b), which offers stronger geometric clues about the scene layout compared to the widely used point mapping (a). Part of the success of our pipeline attributes to the modeling of structural priors such as coincidence (c), and parallelism / orthogonality (d). The corresponding 3D association graphs between lines and points / vanishing points (VPs) are also recovered from our system as a byproduct. The degree-1 point and degree-2 junctions are colored in blue and red respectively in (c), while parallel lines associated with the same VP are colored the same in (d).

where it is difficult to detect and match sufficiently many stable keypoints, such as in indoor areas. On the contrary, these man-made scenes contain abundant lines, e.g. in walls, windows, doors, or ceilings. Furthermore, lines exhibit higher localization accuracy with less uncertainty in pixels [16]. Last but not least, lines appear in highly structured patterns, often satisfying scene-wide geometric constraints such as co-planarity, coincidence (line intersections), parallelism, and orthogonality. In practice, lines suffer from different issues, such as poor endpoint localization and partial occlusion. However, recent line detectors and matchers are bridging the gap of performance between points and lines [25, 46, 84], making it timely to revisit the line reconstruction problem.

Despite their rich geometric properties and abundance in the real world, there exist very few line-based reconstruction methods in the literature [22, 23, 44, 77]. In practical applications, they have also not achieved the same level of success as their point-based counterparts. We believe this is due to several intrinsic challenges specific to line mapping:

- **Inconsistent endpoints.** Due to partial occlusion, lines often have inconsistent endpoints across images.
- **Line fragmentation.** In each image there might be multiple line segments that belong to the same line in 3D. This makes the process of creating track associations more complex compared to building 3D point tracks.
- **No two-view geometric verification.** While point matches can be verified in two views via epipolar geometry, lines require at least three views to filter.
- **Degenerate configurations.** In practice line triangulation is more prone to unstable configurations (see Fig. 8), e.g. becoming degenerate whenever the line is parallel with the camera motion (i.e. to epipolar lines).
- **Weaker descriptor-based matching.** State-of-the-art descriptors for line segments are far behind their point-based counterparts, putting more emphasis on geometric verification and filtering during reconstruction.

In this paper we aim to reduce the gap between point-based and line-based mapping solutions. We propose a new robust mapping method, LIMAP, that integrates seamlessly into existing open-source point-based SfM frameworks [64, 67, 80]. By sharing the code with the research community we hope to enable more research related to lines; both for low-level tasks (such as improving line segment detection and description) and for integrating lines into higher-level tasks (such as visual localization or dense reconstruction). In particular, we make the following contributions in the paper:

- We build a new line mapping system that **reliably reconstructs 3D line segments from multi-view RGB images**. Compared to previous approaches, our line maps are significantly more complete and accurate, while having more robust 2D-3D track associations.
- We achieve this by **automatically identifying and exploiting structural priors** such as coincidence (junctions) and parallelism. **Our technical contribution spans all stages of line mapping** including triangulating proposals, scoring, track building, and joint optimization, with 3D line-point / VP association graphs output as a byproduct.
- The framework is **flexible** such that researchers can easily change components (e.g. detectors, matchers, vanishing point estimators, etc.) or integrate additional sensor data (e.g. depth maps or other 3D information).
- We are the first to go beyond small test sets by quantitatively evaluating on both synthetic and real datasets to benchmark the performance, with hundreds of images for each scene, in which **LIMAP consistently and significantly outperforms existing approaches**.
- Finally, we demonstrate the usefulness of having robust line maps by showing **improvement over purely point-based methods** in tasks such as visual localization and bundle adjustment in Structure-from-Motion.

2. Related Work

Line Detection and Matching. Detecting 2D line segments conventionally relies on grouping image gradients [5, 75]. To improve the robustness and repeatability, learning-based line detectors were later proposed to tackle the problem of wireframe parsing [25, 43, 82, 83, 88, 90]. Recent deep detectors [26, 46, 81] manage to achieve impressive results for detecting general line segments. Matching of the detected line segments is often based on comparing either handcrafted [8, 74, 76, 85] or learning-based [1, 34, 46, 73, 84] descriptors. Some recent methods also exploit point-line [14, 15] and line-junction-line structures [38, 39] to improve matching results, yet still not reaching the reliability level of advanced point matchers [58, 70]. Our method can leverage any line detector and matcher, and is robust to outliers.

Line Reconstruction. As a seminal work, Bartoli and Sturm [6, 7] proposed a full SfM pipeline for line segments, later improved by Schindler [63] with Manhattan-world assumption [12]. Jain et al. [27] proposed to impose global topological constraints between neighboring lines, which were further explored in [51, 53, 54] to build wireframe models. Some learning-based methods [42, 90] were introduced as well to predict 3D wireframes. Hofer et al. [21–23] proposed checking weak epipolar constraints over exhaustive matches and graph clustering, and introduced the Line3D++ software (referred as L3D++ in this paper), which remains the top choice [17, 42] for acquiring 3D line maps so far. Recently, ELSR [77] employed planes and points to guide the matching. However, all prior work mainly shows qualitative results and provides quantitative evaluation only on relatively small image sets [27, 69]. In this paper, we set up a quantitative evaluation on benchmarks with hundreds of images, where our proposed system significantly surpasses prior work by improving all stages in the mapping pipeline.

Line-based Applications. The resulting 3D line maps can be used for many downstream applications. [23] advocates the complementary nature of line reconstruction for structure visualization. Some incremental line-based SfM systems are introduced in [24, 44, 86]. To improve quality and robustness, recent methods [18, 19, 40, 41, 49, 78, 91] jointly employ point and line features in SLAM. While their line maps are often noisy and incomplete, noticeable improvement has been achieved in the accuracy of the recovered camera motion. There has also been development on VP estimation [9, 37, 50, 87] and solvers for joint point-line pose estimation [4, 52, 72, 89]. Recently, promising performance in visual localization has been achieved by combining point and line features in a refinement step [17]. In this paper, we show that our line maps can benefit multiple applications such as localization, SfM, and MVS (Sec. J in supp.). In particular, we present very competitive results on point-line visual localization.

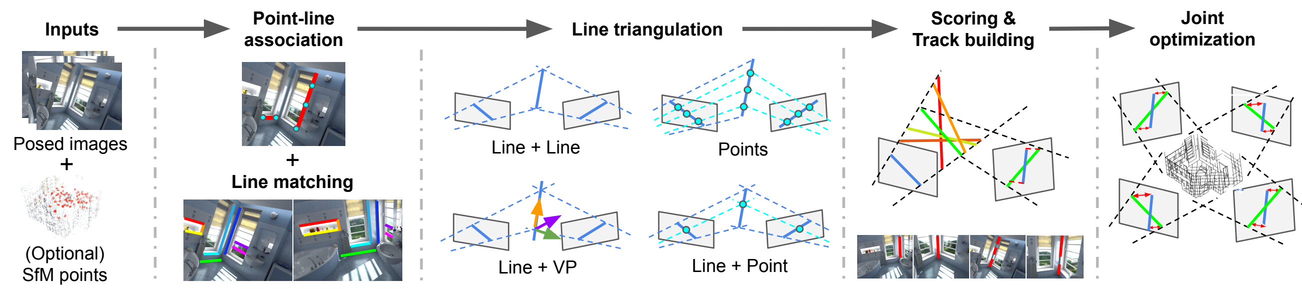


Figure 2. **Overview.** Given a set of posed images and optional 3D points, we associate nearby points to lines, match the lines, triangulate them with 4 different strategies, score 3D line proposals, build line tracks, jointly optimize all features, before obtaining our final reconstruction.

3. The Proposed 3D Line Mapping Pipeline

We now present our proposed pipeline for 3D line mapping. Our method takes as input a set of images with 2D line segments from any existing line detectors. We assume the camera pose for each image is available (e.g. from SfM/SLAM), and optionally we can also leverage a 3D point cloud (e.g. obtained from point-based SfM). The pipeline consists of three main steps:

- **Proposal Generation (Sec. 3.1)**: For each 2D line segment, we generate a set of 3D line segment proposals.
- **Scoring and Track Association (Sec. 3.2)**: Considering multi-view consistency, we score each proposal, select the best candidate for each 2D line, and associate them into a set of 3D line tracks.
- **Joint Refinement (Sec. 3.3)**: Finally, we jointly perform non-linear refinement over the 3D line tracks along with 3D points and VP directions, integrating additional structural priors as soft constraints.

Figure 2 shows an overview of the overall pipeline. In the following sections, we detail each of the three main steps.

By design our pipeline is robust to scale changes and **we use the same hyper-parameters for all experiments across datasets**, which are provided in Sec. F.2 in the supp.

3.1. Generating 3D Line Segment Proposals

The first step is to generate a set of 3D line proposals for each 2D line segment. Given a segment in an image, we use any existing line matcher to retrieve the top K line matches in each of the n_v closest images. Using the top K line matches instead of a single match increases the chance of getting a correct match, while wrong matches will be filtered out in subsequent steps.

Let $(\boldsymbol{x}_1^r, \boldsymbol{x}_2^r) \in \mathbb{R}^3 \times \mathbb{R}^3$ be the two endpoints (in homogeneous coordinates normalized by the intrinsics) for the reference line segment that we wish to generate proposals for. For ease of notation, we let the world-coordinate system align with the reference view. The endpoints of the 3D line proposals that we generate can all be written as

$$\boldsymbol{X}_1 = \lambda_1 \boldsymbol{x}_1^r, \quad \boldsymbol{X}_2 = \lambda_2 \boldsymbol{x}_2^r, \tag{1}$$

for some values of $\lambda_1, \lambda_2 \in \mathbb{R}$. Having the 3D endpoints of all proposals lie on the camera rays of the 2D endpoints simplifies the scoring procedure in the second step (Sec. 3.2).

3.1.1 Line Triangulation

For each matched 2D line segment $(\boldsymbol{x}_1^m, \boldsymbol{x}_2^m)$ we generate one proposal via algebraic line triangulation. Let $(R^m, \boldsymbol{t}^m)$ be the camera pose of the matched view. We can then solve linearly for the endpoint ray depths λ_i as

$$(\boldsymbol{x}_1^m \times \boldsymbol{x}_2^m)^T \left(R^m(\lambda_i \boldsymbol{x}_i^r) + \boldsymbol{t}^m\right) = 0, \quad i = 1, 2. \tag{2}$$

The proposals are then filtered with cheirality checks (positive λ) and degeneracy check via the angle between ray $\boldsymbol{x}_i^r$ and $\boldsymbol{\ell_m} = \boldsymbol{x}_1^m \times \boldsymbol{x}_2^m$. Note that line triangulation becomes inherently unstable close to degenerate configurations when $\boldsymbol{\ell}_m^T R^m \boldsymbol{x}_i^r = 0$, where we get zero or infinite solutions from (2). Geometrically, this happens when the line is parallel with the epipolar plane: If $\boldsymbol{\ell}_m^T \boldsymbol{t}^m \neq 0$ they have no intersection, otherwise they intersect fully and we get infinite solutions $\boldsymbol{\ell}_m \sim \boldsymbol{t}^m \times R^m \boldsymbol{x}_i^r = E\boldsymbol{x}_i^r$, i.e. the line segment coincides with the epipolar line from $\boldsymbol{x}_i^r$. This issue is further illustrated in Figure 8. Since we solve for each λ_i independently, the triangulation problem can have zero, one, or two degenerate endpoints. We term the case with one degenerate endpoint as a *weakly degenerate* one, and the case with two degenerate endpoints as *fully degenerate*. In contrast to the point case, two-view line triangulation is minimal such that any solution fits the measurements exactly with zero error, preventing filtering with 2D reprojection error at this stage.

3.1.2 Point-Line Association

To obtain meaningful proposals in degenerate cases, we leverage additional geometric information coming from either points or associated vanishing points (VPs). 2D-3D point correspondences can either come from a point-based SfM model or be triangulated from matched endpoints/junctions. For each 2D line segment, we associate all 2D points within a fixed pixel threshold and thereby associate with their corresponding 3D points. For each image, we also estimate a set of VPs and their association to 2D lines using JLinkage [71].

3.1.3 Point-guided Line Triangulation

We now generate a second set of proposals for each 2D line segment with the assistance of the associated 2D-3D point correspondences and vanishing points. In the following parts we present three different methods. M1 employs multiple associated 3D points so it is stable for all cases including the *fully degenerate* ones, while M2 and M3 with one known point / VP can help generate stable proposals in *weakly degenerate* cases, which are more common in practice. Cheirality tests are applied to all proposals with respect to both views.

M1. Multiple Points. For each matched line segment we generate one proposal by collecting all of the associated 3D points that are common between the reference and the match. On top of those common points, we fit a 3D line that is then projected onto two camera rays corresponding to $\boldsymbol{x}_1^r$ and $\boldsymbol{x}_2^r$.

M2. Line + Point. For each matched line segment we also generate one proposal for each shared 3D point. We first project the 3D point onto the plane spanned by $\boldsymbol{x}_1^r$ and $\boldsymbol{x}_2^r$. We then aim to find a line that passes through the projection and minimizes the residuals in (2) to the matched line. This can be formulated as a quadratic optimization problem in the two endpoint depths $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ with a single constraint:

$$\min_{\boldsymbol{\lambda} \in \mathbb{R}^2} \boldsymbol{\lambda}^T A \boldsymbol{\lambda} + \boldsymbol{b}^T \boldsymbol{\lambda}, \quad \text{s.t.} \quad \boldsymbol{\lambda}^T Q \boldsymbol{\lambda} + \boldsymbol{q}^T \boldsymbol{\lambda} = 0. \tag{3}$$

Due to the low-dimensionality of the problem, a closed-form solution can be derived by reducing it to a univariate quartic polynomial. We show the full derivation in Sec. B in supp.

M3. Line + VP. Each VP corresponds to a 3D direction. For each associated VP, we generate one proposal based on its direction (again projected onto the plane spanned by $\boldsymbol{x}_1^r$ and $\boldsymbol{x}_2^r$). This gives a single linear constraint on the ray depths,

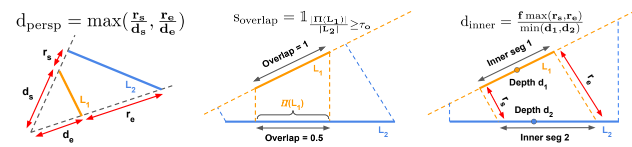
$$(\boldsymbol{v} \times (\boldsymbol{x}_1^r \times \boldsymbol{x}_2^r))^T (\lambda_2 \boldsymbol{x}_2^r - \lambda_1 \boldsymbol{x}_1^r) = 0. \tag{4}$$

where $\boldsymbol{v} \in \mathbb{R}^3$ is the VP. Using the constraint, we then solve for $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ by minimizing the two residuals of (2) in a least squares sense. Note that $\boldsymbol{v}$ can either come from the reference image, or from a matched line in another image.

Extension: Line Mapping Given Depth Maps. The proposal generation step can be improved when each image has a corresponding depth map (e.g. from an RGB-D sensor), which can be leveraged with robust line fitting to generate the 3D line proposals. Refer to Sec. E in our supplementary material for more details and results.

3.2. Proposal Scoring and Track Association

At this point, each 2D line segment l in image I is associated with a set $\mathcal{K}$ of 3D line segment proposals (stemming from the top K line matches and various triangulations) for each neighboring image J. We describe in the following how we select the best 3D line proposal for each 2D line segment, and associate these lines into tracks. For each of these



(a) Perspective distance (b) Overlap score (c) InnerSeg distance

Figure 3. **Scoring methods.** We propose three novel line scoring measures that are scale-invariant and handle different line lengths.

steps, we leverage different scoring methods quantifying the distance between two 3D line segments (L_1, L_2). These distances are usually computed symmetrically and averaged, and can be obtained both in 3D and in 2D by projecting each 3D line into the other view. We start by presenting two classic ones, and then define our three novel line distances (one for 3D proposal selection and two for track building).

- *Angular distance*: angle between L_1 and L_2.
- *Perpendicular distance*: maximum orthogonal distance of the endpoints of L_1 to the infinite line spanned by L_2.

3D Proposal Selection. To select best 3D candidate for each 2D line, we score each proposal L_i by measuring its consistency with the others. Here we introduce a new distance:

- *Perspective distance*: assuming the endpoints of L_1 and L_2 are on the same rays as in Fig. 3(a), the distance is defined as the endpoint distances, divided by the ray depths d_s, d_e of the endpoints of L_1 in image 1. This score can filter out ill-posed triangulations (refer to Sec. F.3 in supp. for detailed discussions), while remaining scale-invariant.

This new distance, together with the *angular distance* in 2D and 3D, and the *perpendicular distance* in 2D, have different scales. In order to aggregate them together, we associate a scaling factor τ_r to each distance r and get a normalized score $s_n = e^{-(r/\tau_r)^2} \in (0, 1]$. Denoting by $\mathcal{S}$ the set of all the corresponding normalized scores and $\mathbb{1}$ the indicator function, the score between L_1 and L_2 becomes

$$s(L_1, L_2) = \min_{s_n \in \mathcal{S}} (s_n \cdot \mathbb{1}_{s_n \geq 0.5}) \in \{0\} \cup [0.5, 1]. \tag{5}$$

Now equipped with unique score per line pair, we can consider all the neighboring 3D line candidates L_J^k coming from the neighboring image J and proposal k. The consistency score is defined by summing the best score from each image:

$$s_c(L_i) = \sum_{J \in \mathcal{N}_I} \max_{k \in \mathcal{K}} s(L_i, L_J^k), \tag{6}$$

where $\mathcal{N}_I$ is the set of neighboring images of I. The best 3D line candidate for each 2D line segment l is then selected as the proposal with the highest score: $L = \text{argmax}_{L_i}\, s_c(L_i)$. If the score is less than 1.0, i.e. the best candidate has less than two supports from neighboring views, we ignore this 2D line segment in the subsequent track building process.

Track Building. At this point, each 2D segment has been assigned a unique 3D line (its best 3D line candidate). The goal of this step is to gather these 2D segments into line tracks. For this, we form a graph where the 2D segments are nodes and all initial line matches are edges. We aim to prune edges in the graph such that the connected 2D segments share similar 3D assignments. We propose two new line scoring measures that can cope with different endpoint configurations and variable scales across images.

- *Overlap score*: we project L_1 orthogonally onto L_2, clip the projected endpoints to the endpoints of L_2 if they fall outside of L_2 to get segment $\Pi(L_1)$, and compare the ratio of lengths to a threshold τ_o: $\mathbb{1}_{\frac{|\Pi(L_1)|}{|L_2|} \geq \tau_o}$ (see Fig. 3(b)).
- *InnerSeg distance*: the endpoints of L_1 are perpendicularly unprojected to L_2. If they fall outside of L_2, we clip them to the closest endpoint of L_2. By doing this in both directions, we can define two *inner segments* (see Fig. 3(c)), and the *InnerSeg* distance as the maximum distance between their endpoints. To make this measure scale-invariant, we additionally divide it by a scale factor $\sigma = \frac{\min(d_1, d_2)}{f}$, where d_j is the depth of the mid-point of L_j in image J and f is the focal length. This encodes how far the mid-point can move in 3D before reaching 1 pixel error in the image (detailed in Sec. F.3 in supp.).

We then convert the *InnerSeg distance* computed in 3D to a normalized score as in the previous paragraph, and combine it with the *overlap score* in 2D and 3D and previous scores using (5). Given these pairwise scores of 3D lines, we can now prune edges whose score is below a threshold $t_f = 0.5$. The connected components of the resulting graph yield the line tracks, ignoring components with less than 3 nodes.

For each track, we then re-estimate a single 3D line segment. Using the set of endpoints from the 3D assignments of all nodes in the track, we apply Principal Component Analysis (PCA) and use the principal eigenvector and mean 3D point to estimate the infinite 3D line. We then project all endpoints on this infinite line to get the new 3D endpoints.

3.3. Joint Optimization of Lines and Structures

Finally, we perform non-linear refinement on the acquired 3D lines with their track information. The straightforward approach is to perform geometric refinement on the reprojection error. With the 2D point-line association available, we can formulate a joint optimization problem by including additional structural information. The energy to minimize can be written as follows:

$$E = \sum_p E_P(p) + \sum_l E_L(l) + \sum_{(p,l)} E_{PL}(p,l), \quad (7)$$

where E_P and E_L are the data terms, and E_{PL} encodes the 3D association between lines and points / VPs. In particular, E_P is the 2D point reprojection error as in regular bundle adjustment [64]. The association energy is softly weighted (as discussed later) and optimized with robust Huber loss [3]. Each line is converted into a 4-DoF infinite line with Plücker coordinate [7] for optimization and converted back to line segments by unprojecting its 2D supports. Each vanishing point is parameterized with a 3-dimensional homogeneous vector. Refer to Sec. A in supp. for details on efficient computation with minimal parameterization.

Geometric Refinement. The data term of each line track is also defined on its 2D reprojections. In particular, we measure the 2D perpendicular distance weighted by the angle consistency, which we robustly equip with Cauchy loss [3]:

$$E_L(l) = \sum_k w_\angle^2(L_k, \ell_k) \cdot e_{\text{perp}}^2(L_k, \ell_k), \quad (8)$$

where e_{perp} is the perpendicular distance, L_k is the 2D projection of the 3D segment, ℓ_k are the 2D line segments, and $w_\angle$ is the exponential of one minus the cosine of the 2D angle between the projected and the observed line.

Soft Association between Lines and Points. For each pair of 3D line and 3D point with their track information, we can estimate how likely they are spatially associated by traversing the 2D association graph (described in Sec. 3.1.2) of their supports. Specifically, we count the number of associations among the 2D supports of the line track and point track, and keep pairs with at least three 2D associations. The 3D association energy E_{PL}, defined on the surviving pairs, is formulated as the 3D point-line distance weighted by the number of 2D associations on their supports.

Soft Association between Lines and VPs. Same as the point case, we can also build a soft association problem between lines and VPs. First, we acquire 3D VP tracks by transitively propagating line correspondences from the 3D line tracks. Then, we count the number of associations among the 2D supports for each pair of 3D line and VP track. The 3D line-VP association energy is defined as the sine of the direction angle between the 3D line and the VP, implicitly enforcing parallelism. Furthermore, we add regularizations to the nearly orthogonal VP pairs to enforce orthogonality of different line groups. Refer to Sec. C in supp. for details.

4. Experiments

Implementation Details. Our whole library is implemented in C++ with Python bindings [28]. The triangulation and scoring can be run in parallel for each node, enabling scalability to large datasets. We use $n_v = 20$ visual neighbors and keep the top $K = 10$ line matches. We provide all the values of thresholds and scaling factors in Sec. F.2 in supp.

4.1. Line Mapping

To validate the effectiveness of our system, we set up an evaluation benchmark to quantify the quality of the recon-

Line type	Method	R1	R5	R10	P1	P5	P10	# supports
LSD [75]	L3D++ [23]	37.0	153.1	218.8	53.1	80.8	**90.6**	(14.8 / 16.8)
	ELSR [77]	13.9	59.7	96.5	55.4	72.6	82.2	(N/A / N/A)
	Ours	**48.6**	**185.2**	**251.3**	**60.1**	**82.4**	90.0	(**16.4** / **20.5**)
SOLD2 [46]	L3D++ [23]	36.9	107.5	132.8	67.2	**86.8**	**93.2**	(13.2 / 20.4)
	Ours	**54.3**	**151.1**	**191.2**	**69.8**	84.6	90.0	(**16.5** / **38.7**)

Table 1. **Line reconstruction on Hypersim [55]** with LSD [75] and SOLD2 [46] lines. $R\tau$ and $P\tau$ are reported at 1mm, 5mm, 10 mm along with the average number of supporting images/lines.

Method	R5	R10	R50	P5	P10	P50	# supports
L3D++ [23]	373.7	831.6	2783.6	40.6	54.5	85.9	(8.8 / 9.3)
ELSR [77]	139.2	322.5	1308.0	38.5	48.0	74.5	(N/A / N/A)
Ours (line-only)	472.1	1058.8	3720.7	**46.8**	**58.4**	**86.1**	(10.3 / 11.8)
Ours	**508.3**	**1154.5**	**4179.5**	46.0	56.9	83.7	(**10.4** / **12.0**)

Table 2. **Line reconstruction on *train* split of *Tanks and Temples* [32]** with LSD [75] lines. $R\tau$ and $P\tau$ are reported at 5mm, 10mm, 50mm along with the average number of supporting images/lines.

structed 3D line maps. As there are no ground truth (GT) 3D lines, we evaluate the 3D line mapping with either GT mesh models or point clouds. We use the following metrics:

- *Length recall* (in meters) at τ ($R\tau$): sum of the lengths of the line portions within τ mm from the GT model.
- *Inlier percentage* at τ ($P\tau$): the percentage of tracks that are within τ mm from the GT model.
- *Average supports*: average number of image supports and 2D line supports across all line tracks.

In the following, we compare our system with two state-of-the-art methods as baselines: L3D++ [23] and ELSR [77], using two line detectors: the traditional LSD detector [75] and the learning-based SOLD2 [46]. For ELSR [77], we convert the input into VisualSfM [80] format and use code[1] from the authors (only supporting LSD [75]).

Our first evaluation is run on the first eight scenes of the Hypersim dataset [55], composed of 100 images each, and is reported in Tab. 1. For both detectors, we reconstruct much more complete line maps with better or comparable precision than the competitors, while also exhibiting significantly higher quality of track information. This abundant track association is beneficial particularly for line-based applications such as visual localization [17]. After discussing with the authors of ELSR, it seems that their method does not achieve satisfactory results due to a lack of point and plane features.

We further evaluate all three methods on the *train* split of the *Tanks and Temples* dataset [32] without *Ignatius* as it has no line structures. As SOLD2 [46] is trained for indoor images, we only use LSD [75]. Since the provided point cloud was cleaned to focus only on the main subject, we compute its bounding box, extend it by one meter, and only evaluate lines inside this region. This prevents incorrectly penalizing correct lines that are far away from the main scene,

[1] https://skyearth.org/publication/project/ELSR/

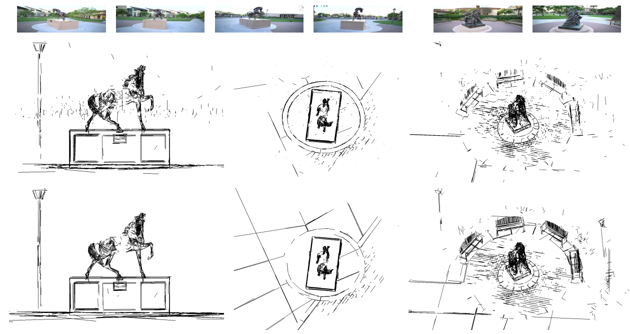
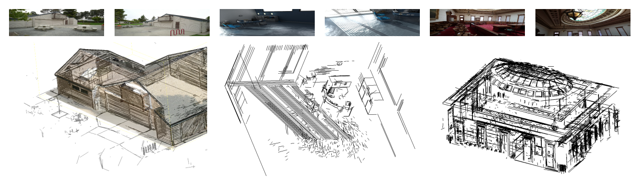


Figure 4. **Top row:** L3D++ [23]. **Bottom row:** Ours. Both systems are run on *Horse* and *Family* from [32]. We show two different views on the main scene of *Horse*.



Barn (410 images) *ai_001_003* (100 images) *Courtroom* (301 images)

Figure 5. **Qualitative results on Hypersim [55] and *Tanks and Temples* [32].** On *Barn* we jointly visualize our results and the aligned ground truth point cloud.

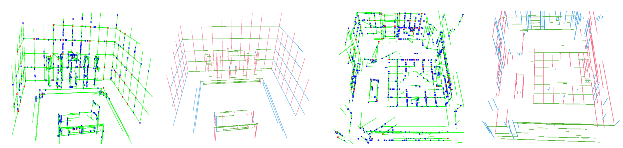


Figure 6. **Qualitative results** of the recovered line-point and line-VP association graphs (visualized similarly as in Fig. 1).

which our method is particularly good at thanks to our scale-invariant design (refer to Sec. G in supp.). Tab. 2 shows the results, where our methods significantly improve the mapping quality across the board. Fig. 4 shows qualitative comparison between our method and L3D++ [23]. Our results exhibit better completeness, have less noisy lines that are flying around, and achieve significantly more robust reconstructions of subtle details (e.g. on the ground). More examples of our produced line maps are shown in Fig. 5.

As an additional output of our system, junction structures and line-line relations such as parallelism and orthogonality are discovered, as shown in Fig. 6. This directly comes from the line-point and line-VP soft associations of Sec. 3.3. From the recovered structures, we can clearly perceive the scene and easily recognize the main Manhattan directions [12].

To demonstrate the scalability of the proposed system, we also run our method on two large-scale datasets: Aachen (6,697 images) [61, 62] and Rome city (16,179 images) [2, 67, 68]. Fig. 7 shows that our method produces reliable line maps with clear structures. Note that the camera poses from Bundler [67] on Rome city are far from perfect, while our mapping still works reasonably well. The efficiency

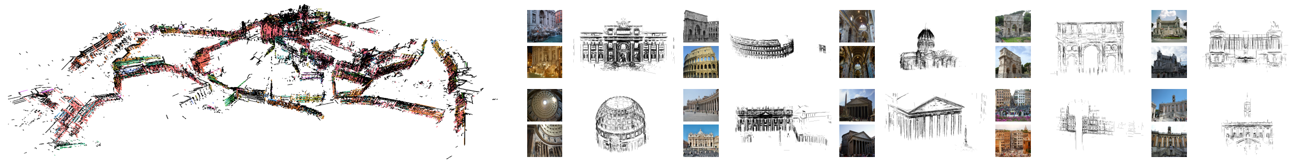


Figure 7. **Scalability to large-scale datasets:** Aachen (6,697 images) [61] and Rome (16,179 images) [2, 67, 68]. For Aachen [61], parallel lines from the line-VP association graph are colored the same. For Rome [2, 67, 68], we visualize 10 representative components individually.

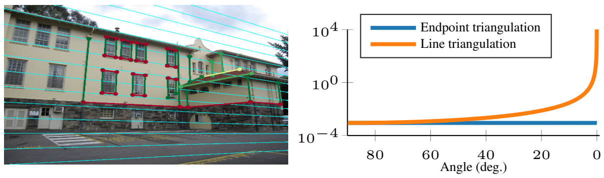


Figure 8. **Uncertainty in line triangulation** measured by the largest eigenvalue of the covariance (Sec. D in supp.). **Left:** Each segment is colored by the uncertainty in the triangulation. Lines that align with the epipolar lines (shown in blue) exhibit higher (red) uncertainty. **Right:** We perform a small synthetic experiment to illustrate this. The graph shows the uncertainty for line triangulation as the lines approach the degenerate state. We compare with point-based triangulation assuming that endpoints are consistent.

Line type	Triangulation	R1	R5	R10	P1	P5	P10	# supports
LSD [75]	Endpoints	27.6	101.4	138.0	58.2	**83.5**	**92.1**	(13.0 / 13.2)
	Line	**48.3**	**187.0**	**257.4**	**59.2**	81.9	89.8	**(15.8 / 19.1)**
SOLD2 [46]	Endpoints	27.3	82.8	106.5	68.2	84.5	90.9	(12.3 / 19.9)
	Line	**50.8**	**143.5**	**180.8**	**74.4**	**86.9**	**91.2**	**(15.1 / 32.2)**

Table 3. **Comparison between endpoint and line triangulation** on Hypersim [55]. While being more stable at triangulation, the endpoints are often unmatched between line pairs.

Line	M1	M2	M3	R1	R5	R10	P1	P5	P10	# supports
✓				50.8	143.5	180.8	74.4	86.9	91.2	(15.1 / 32.2)
	✓			24.9	72.5	95.8	65.9	81.2	88.5	(11.3 / 15.7)
	✓	✓		37.7	116.8	152.6	71.0	84.2	89.7	(13.8 / 25.8)
✓	✓			51.5	146.9	185.4	71.7	85.4	90.1	(14.9 / 31.2)
✓	✓	✓		51.3	146.4	186.4	73.4	85.7	90.5	(15.8 / 35.6)
✓	✓	✓	✓	51.4	145.4	184.9	74.1	86.1	90.6	(16.5 / 38.7)

Table 4. **Ablation study** on different types of triangulation proposals (defined in Sec. 3.1.3) on Hypersim [55] with SOLD2 [46].

bottleneck is in line detection and matching (we use SOLD2 [46] descriptors), while the rest of the mapping takes only ∼10 minutes on Aachen [61, 62]. The time complexity of our system is nearly linear with the number of images.

4.2. More Insights and Ablation Studies

Line Triangulation. To study the stability of the triangulation, we perform a small test on a stereo pair from AdelaideRMF [79] on the uncertainty (measured by the largest singular value of the covariance) of the triangulated 3D segments. We further run a synthetic experiment by generating random lines on a plane orthogonal to the stereo pair, and plot the uncertainty of point and line triangulations with

Line type	Method	R1	R5	P1	P5	# supports
LSD [75]	L3D++ [23]	37.0	153.1	53.1	80.8	(14.8 / 16.8)
	Ours (line) w/ [23] scoring	**48.6**	186.0	56.5	80.6	(14.4 / 16.8)
	Ours (line) w/ [23] merging	41.2	158.2	**59.6**	**82.5**	(15.6 / 16.7)
	Ours (line) w/ exhaustive	46.7	177.2	57.6	80.9	**(16.8 / 20.8)**
	Ours (line)	48.3	**187.0**	59.2	81.9	(15.8 / 19.1)
SOLD2 [46]	L3D++ [23]	36.9	107.5	67.2	86.8	(13.2 / 20.4)
	Ours (line) w/ [23] scoring	45.8	133.2	72.6	85.9	(15.0 / 31.1)
	Ours (line) w/ [23] merging	37.7	113.4	70.5	84.5	(13.3 / 23.9)
	Ours (line) w/ exhaustive	48.9	139.7	72.9	85.7	**(16.2 / 36.9)**
	Ours (line)	**50.8**	**143.5**	**74.4**	**86.9**	(15.1 / 32.2)

Table 5. **Studies on different components of our method** with only line-line proposals against L3D++ [23].

respect to the angle of the lines with the baseline (refer to Sec. D in supp. for details). The results in Fig. 8 show that when the matched line is nearly parallel to the epipolar line, the line triangulation becomes degenerate with exploding uncertainty, while triangulating the endpoints is significantly more stable. Thus, combining points and VPs from the 2D association is beneficial to improve the stability of the proposals. However, the endpoints are generally not consistent across line matches in practice and need to be complemented with line-line triangulation. This can be verified in Tab. 3 where the performance significantly drops when we change line triangulation into endpoint triangulation.

We further ablate our four types of triangulation for generating proposals. Results in Tab. 4 show that integrating points and VPs enhance the 3D line maps, in particular significantly improving the track quality. Another surprising fact is that the third line in the table, relying only on points and line + point triangulation, already achieves better results than the prior baselines in Tab. 1. Employing all four types of proposals obtains the best trade-off.

Scoring and Track Building. We first study the effects of using exhaustive line matching as in L3D++ [23]. To enable direct comparison we only use line triangulation proposals. Results are shown in Tab. 5. While there are more proposals generated from the exhaustive matches, both the recall and precision decrease by a noticeable margin. This is probably due to the large number of wrong proposals misleading the scoring process. Nevertheless, our method with exhaustive matches still works significantly better than L3D++ [23]. To further study the effects of the proposed distance measurements at scoring and track building (merging), we re-implement the ones proposed in L3D++ [23] and perform direct comparison. Both our scoring and track

Method	R1	R5	R10	P1	P5	P10	# supports
Line-only w/o refine	43.5	135.8	180.1	75.1	87.2	92.2	(15.1 / 32.2)
Line-only w/ geom alone	50.8	143.5	180.8	74.4	86.9	91.2	(15.1 / 32.2)
w/o refine	46.5	146.0	189.7	**76.8**	**88.9**	**93.3**	(**16.5 / 38.7**)
w/ geom alone	51.4	145.4	184.9	74.1	86.1	90.6	(**16.5 / 38.7**)
w/ joint optimization	**54.3**	**151.1**	**191.2**	69.8	84.6	90.0	(**16.5 / 38.7**)

Table 6. **Line refinement** on Hypersim [55] with SOLD2 [46].

Dataset	HLoc[2] [56,57]	PtLine [17]	Ours
Cambridge [30]	7.0 / 0.13 / 44.0	7.4 / 0.13 / 43.5	**6.7 / 0.12 / 46.1**
7Scenes [66]	3.3 / 1.08 / 73.0	3.3 / 1.09 / 72.7	**3.0 / 1.00 / 78.0**

Table 7. **Visual localization on Cambridge [31] and 7Scenes [66]**. We report the median translation and rotation errors in cm and degrees, and the pose accuracy (%) at 5 cm / 5 deg threshold. All metrics are averaged across all scenes of each dataset.

HLoc [56,57] Ours w/ LIMAP

	(T / R) err. ↓	Acc. ↑
HLoc [57]	5.2 / 1.46	46.8
HLoc [57] w/ depth	4.7 / 1.25	53.4
PtLine [17]	4.8 / 1.33	51.9
Ours w/ L3D++ [23]	4.1 / 1.14	60.8
Ours w/ LIMAP	**3.7 / 1.02**	**71.1**

Figure 9. **Line-assisted Visual localization** on *Stairs* from 7Scenes [66]. Blue: 2D points/lines; Green/Red: Projected 3D points/lines.

building are significantly better, especially when equipped with SOLD2 [46] which produces more structured lines.

Joint Optimization. Finally, we ablate the proposed joint optimization in our pipeline. First, we remove the point-line association and only apply the geometric residuals (reprojection error). Results in Tab. 6 show that the geometric refinement improves significantly when the proposals solely come from line triangulation. However, when adding additional proposals from points and VPs, it contributes marginally and even misleads some lines that are generated from points and VPs but poorly conditioned for lines (R10 decreases). When integrated with joint optimization with soft association, the recall is further improved noticeably, while sacrificing a bit on the precision. It is worth pointing out that the joint optimization also enables the byproduct of junction structures and line-line relations (e.g. in Fig. 6).

4.3. Applications

Line-Assisted Visual Localization. We build a hybrid visual localization with both points and lines on top of the acquired 3D line maps. Specifically, we first build point maps as in HLoc [56, 57] and line maps with our proposed method. Then, we match points and lines respectively and get 2D-3D correspondences from the track information in the 3D maps. Given these correspondences, we combine four minimal solvers [33, 47, 89]: P3P, P2P1LL, P1P2LL, P3LL from PoseLib [35], together in a hybrid RANSAC framework [10, 59] with local optimization [11, 36] to get the final 6-DoF pose (refer to Sec. H in supp. for details). This also enables direct comparison since only using P3P [47] corresponds to the point-alone baseline similar to HLoc [56, 57]. We also compare with the post-refinement of PtLine [17] that optimizes over the initial point-alone predictions.

[2]Up to the date of submission, the COLMAP model [64] used by HLoc [56, 57] does not consider radial distortion from the VisualSfM [80] model. So our results are better than the original ones.

	Med. error ↓	AUC @ (1° / 3° / 5°) ↑
COLMAP [64]	0.188	77.3 / 89.0 / 91.6
COLMAP [64] + LIMAP refinement	**0.146**	**82.9 / 91.2 / 93.0**

Table 8. **Joint bundle adjustment of points and lines** on Hypersim [55]. Relative pose errors are measured on all image pairs.

Results in Tab. 7 show that our localization system achieves consistently better results than the point-alone baseline both indoors [66] and outdoors [30], validating the effectiveness of employing 3D line maps for visual localization. In Fig. 9 we show more detailed results from the *Stairs* scene from 7Scenes [66] as it is one of the most challenging ones. Integrating lines significantly benefits the alignment of the reprojected structures, improving the pose accuracy from 46.8 to 71.1. Also, with our localization pipeline, using the map built from our proposed method is better than from L3D++ [23] by a noticeable margin, again demonstrating the advantages of our proposed line mapping system.

Refining Structure-from-Motion. With the acquired 3D line maps built from a roughly correct point-based structure-from-motion model, e.g, COLMAP [64], we can use the 3D lines with their track information to refine the input camera poses with joint optimization of points and lines. To verify this, we run COLMAP [64] with SuperPoint [13] on the first eight scenes of Hypersim [55], run the proposed line mapping on top of it, and perform joint bundle adjustment to refine poses and intrinsics. We report the relative pose evaluation of all image pairs [29]. Tab. 8 shows that the joint point-line refinement consistently benefits the accuracy of the camera poses, in particular improving AUC@1° by 5.6.

5. Conclusion

In this paper, we introduce LIMAP: a library for robust 3D line mapping from multi-view imagery. Extensive experiments show that our method, by improving all stages of the reconstruction pipeline, produces significantly more complete 3D lines, with much higher quality of track association. As a byproduct, the method can also recover 3D association graphs between lines and points / VPs. We further show the usefulness of 3D line maps on visual localization and bundle adjustment. Future directions include incremental / real-time structure mapping, distinguishing structural lines from textural lines for wireframe modeling, and exploiting higher-level structures and relations for downstream applications.

Acknowledgements. V. Larsson was supported by ELLIIT.

References

[1] Hichem Abdellali, Robert Frohlich, Viktor Vilagos, and Zoltan Kato. L2d2: Learnable line detector and descriptor. In *3DV*, 2021. 2

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 6, 7

[3] Sameer Agarwal and Keir Mierle. Ceres solver. `http://ceres-solver.org`. 5

[4] Sérgio Agostinho, João Gomes, and Alessio Del Bue. Cvxpnpl: A unified convex solution to the absolute pose estimation problem from point and line correspondences. *arXiv preprint arXiv:1907.10545*, 2019. 2

[5] Cuneyt Akinlar and Cihan Topal. Edlines: Real-time line segment detection by edge drawing (ed). In *IEEE International Conference on Image Processing*, 2011. 2

[6] Adrien Bartoli, Mathieu Coquerelle, and Peter Sturm. A framework for pencil-of-points structure-from-motion. In *ECCV*, 2004. 2

[7] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding (CVIU)*, 100(3):416–441, 2005. 2, 5

[8] Herbert Bay, Vittorio Ferraris, and Luc Van Gool. Wide-baseline stereo matching with line segments. In *CVPR*, 2005. 2

[9] Jean-Charles Bazin, Yongduek Seo, Cédric Demonceaux, Pascal Vasseur, Katsushi Ikeuchi, Inso Kweon, and Marc Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *CVPR*, 2012. 2

[10] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid camera pose estimation. In *CVPR*, 2018. 8

[11] Ondrej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. In *Joint Pattern Recognition Symposium*, pages 236–243, 2003. 8

[12] James Coughlan and Alan L Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *NeurIPS*, 2000. 2, 6

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 1, 8

[14] Bin Fan, Fuchao Wu, and Zhanyi Hu. Line matching leveraged by point correspondences. In *CVPR*, 2010. 2

[15] Bin Fan, Fuchao Wu, and Zhanyi Hu. Robust line matching through line–point invariants. *Pattern Recognition*, 45(2):794–805, 2012. 2

[16] Wolfgang Förstner and Bernhard P Wrobel. *Photogrammetric computer vision*. Springer, 2016. 1

[17] 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 *IROS*, 2022. 2, 6, 8

[18] Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuniga-Noël, Davide Scaramuzza, and Javier Gonzalez-Jimenez. Pl-slam: A stereo slam system through the combination of points and line segments. *IEEE Transactions on Robotics*, 35(3):734–746, 2019. 2

[19] Yijia He, Ji Zhao, Yue Guo, Wenhao He, and Kui Yuan. Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features. *Sensors*, 18(4):1159, 2018. 2

[20] Jared Heinly, Johannes L. Schönberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world in six days. In *CVPR*, 2015. 1

[21] Manuel Hofer, Michael Maurer, and Horst Bischof. Improving sparse 3d models for man-made environments using line-based 3d reconstruction. In *3DV*, 2014. 2

[22] Manuel Hofer, Michael Maurer, and Horst Bischof. Line3d: Efficient 3d scene abstraction for the built environment. In *German Conference on Pattern Recognition*, 2015. 1, 2

[23] Manuel Hofer, Michael Maurer, and Horst Bischof. Efficient 3d scene abstraction using line segments. *Computer Vision and Image Understanding (CVIU)*, 157:167–178, 2017. 1, 2, 6, 7, 8

[24] Aleksander Holynski, David Geraghty, Jan-Michael Frahm, Chris Sweeney, and Richard Szeliski. Reducing drift in structure from motion using extended features. In *3DV*, 2020. 2

[25] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *CVPR*, 2018. 1, 2

[26] Siyu Huang, Fangbo Qin, Pengfei Xiong, Ning Ding, Yijia He, and Xiao Liu. Tp-lsd: Tri-points based line segment detector. In *ECCV*, 2020. 2

[27] Arjun Jain, Christian Kurz, Thorsten Thormählen, and Hans-Peter Seidel. Exploiting global connectivity constraints for reconstruction of 3d line segments from images. In *CVPR*, 2010. 2

[28] Wenzel Jakob, Jason Rhinelander, and Dean Moldovan. pybind11 – seamless operability between c++11 and python. `https://github.com/pybind/pybind11`. 5

[29] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *IJCV*, 129(2):517–547, 2021. 8

[30] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 8

[31] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In *ICCV*, 2015. 8

[32] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 6

[33] Zuzana Kukelova, Jan Heller, and Andrew Fitzgibbon. Efficient intersection of three quadrics and applications in computer vision. In *CVPR*, 2016. 8

[34] Manuel Lange, Fabian Schweinfurth, and Andreas Schilling. Dld: A deep learning based line descriptor for line feature matching. In *IROS*, 2019. 2

[35] Viktor Larsson. PoseLib - Minimal Solvers for Camera Pose Estimation. https://github.com/vlarsson/PoseLib. 8
[36] Karel Lebeda, Jiri Matas, and Ondrej Chum. Fixing the Locally Optimized RANSAC. In *BMVC*, 2012. 8
[37] Haoang Li, Ji Zhao, Jean-Charles Bazin, Wen Chen, Zhe Liu, and Yun-Hui Liu. Quasi-globally optimal and efficient vanishing point estimation in manhattan world. In *ICCV*, 2019. 2
[38] Kai Li, Jian Yao, and Xiaohu Lu. Robust line matching based on ray-point-ray structure descriptor. In *ACCV*, 2014. 2
[39] 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. 2
[40] Hyunjun Lim, Jinwoo Jeon, and Hyun Myung. Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping. *IEEE Robotics and Automation Letters*, 7(2):1518–1525, 2022. 2
[41] Hyunjun Lim, Yeeun Kim, Kwangik Jung, Sumin Hu, and Hyun Myung. Avoiding degeneracy for monocular visual slam with point and line features. In *ICRA*, 2021. 2
[42] Yicheng Luo, Jing Ren, Xuefei Zhe, Di Kang, Yajing Xu, Peter Wonka, and Linchao Bao. Lc2wf:learning to construct 3d building wireframes from 3d line clouds. In *BMVC*, 2022. 2
[43] Quan Meng, Jiakai Zhang, Qiang Hu, Xuming He, and Jingyi Yu. Lgnn: A context-aware line segment detector. In *ACM International Conference on Multimedia*, 2020. 2
[44] Branislav Micusik and Horst Wildenauer. Structure from motion with line segments under relaxed endpoint constraints. *IJCV*, 124(1):65–79, 2017. 1, 2
[45] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1
[46] Rémi Pautrat, Juan-Ting Lin, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Sold2: Self-supervised occlusion-aware line description and detection. In *CVPR*, 2021. 1, 2, 6, 7, 8
[47] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *ECCV*, 2018. 8
[48] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *CVPR*, 2019. 1
[49] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer. Pl-slam: Real-time monocular visual slam with points and lines. In *ICRA*, 2017. 2
[50] Yiming Qian and James H. Elder. A reliable online method for joint estimation of focal length and camera rotation. In *ECCV*, 2022. 2
[51] Srikumar Ramalingam, Michel Antunes, Dan Snow, Gim Hee Lee, and Sudeep Pillai. Line-sweep: Cross-ratio for wide-baseline matching and 3d reconstruction. In *CVPR*, 2015. 2
[52] Srikumar Ramalingam, Sofien Bouaziz, and Peter Sturm. Pose estimation using both points and lines for geo-localization. In *ICRA*, 2011. 2
[53] Siddhant Ranade and Srikumar Ramalingam. Novel single view constraints for manhattan 3d line reconstruction. In *3DV*, 2018. 2
[54] Jing Ren, Biao Zhang, Bojian Wu, Jianqiang Huang, Lubin Fan, Maks Ovsjanikov, and Peter Wonka. Intuitive and efficient roof modeling for reconstruction and synthesis. In *ACM SIGGRAPH Asia*, 2021. 2
[55] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021. 6, 7, 8
[56] Paul-Edouard Sarlin. Visual localization made easy with hloc. https://github.com/cvg/Hierarchical-Localization/. 8
[57] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 8
[58] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2
[59] Torsten Sattler et al. RansacLib - A Template-based *SAC Implementation. https://github.com/tsattler/RansacLib. 8
[60] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, 2011. 1
[61] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018. 6, 7
[62] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 6, 7
[63] Grant Schindler, Panchapagesan Krishnamurthy, and Frank Dellaert. Line-based structure from motion for urban environments. In *International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2006. 2
[64] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 5, 8
[65] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1
[66] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 8
[67] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH*, 2006. 2, 6, 7
[68] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2):189–210, 2008. 6, 7

[69] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008. 2

[70] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2

[71] Roberto Toldo and Andrea Fusiello. Robust multiple structures estimation with j-linkage. In *ECCV*, 2008. 3

[72] Alexander Vakhitov, Jan Funke, and Francesc Moreno-Noguer. Accurate and linear time pose estimation from points and lines. In *ECCV*, 2016. 2

[73] Alexander Vakhitov and Victor Lempitsky. Learnable line segment descriptor for visual slam. *IEEE Access*, 7:39923–39934, 2019. 2

[74] Bart Verhagen, Radu Timofte, and Luc Van Gool. Scale-invariant line descriptors for wide baseline matching. In *WACV*, 2014. 2

[75] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *TPAMI*, 32(4):722–732, 2008. 2, 6, 7

[76] Zhiheng Wang, Fuchao Wu, and Zhanyi Hu. Msld: A robust descriptor for line matching. *Pattern Recognition*, 42(5):941–953, 2009. 2

[77] Dong Wei, Yi Wan, Yongjun Zhang, Xinyi Liu, Bin Zhang, and Xiqi Wang. Elsr: Efficient line segment reconstruction with planes and points guidance. In *CVPR*, 2022. 1, 2, 6

[78] Xinyu Wei, Jun Huang, and Xiaoyuan Ma. Real-time monocular visual slam by combining points and lines. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2019. 2

[79] Hoi Sim Wong, Tat-Jun Chin, Jin Yu, and David Suter. Dynamic and hierarchical multi-structure geometric model fitting. In *ICCV*, 2011. 7

[80] Changchang Wu. Visualsfm: A visual structure from motion system. *http://www. cs. washington. edu/homes/ccwu/vsfm*, 2011. 2, 6, 8

[81] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *CVPR*, 2021. 2

[82] Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, and Liangpei Zhang. Learning attraction field representation for robust line segment detection. In *CVPR*, 2019. 2

[83] Nan Xue, Tianfu Wu, Song Bai, Fudong Wang, Gui-Song Xia, Liangpei Zhang, and Philip HS Torr. Holistically-attracted wireframe parsing. In *CVPR*, 2020. 2

[84] 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. 1, 2

[85] 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. 2

[86] Lilian Zhang and Reinhard Koch. Structure and motion from line correspondences: Representation, projection, initialization and sparse bundle adjustment. *Journal of Visual Communication and Image Representation*, 25(5):904–915, 2014. 2

[87] Lilian Zhang, Huimin Lu, Xiaoping Hu, and Reinhard Koch. Vanishing point estimation and line classification in a manhattan world with a unifying camera model. *IJCV*, 117, 2015. 2

[88] Ziheng Zhang, Zhengxin Li, Ning Bi, Jia Zheng, Jinlei Wang, Kun Huang, Weixin Luo, Yanyu Xu, and Shenghua Gao. Ppgnet: Learning point-pair graph for line segment detection. In *CVPR*, 2019. 2

[89] Lipu Zhou, Jiamin Ye, and Michael Kaess. A stable algebraic camera pose estimation for minimal configurations of 2d/3d point and line correspondences. In *ACCV*, 2018. 2, 8

[90] Yichao Zhou, Haozhi Qi, Yuexiang Zhai, Qi Sun, Zhili Chen, Li-Yi Wei, and Yi Ma. Learning to reconstruct 3d manhattan wireframes from a single image. In *ICCV*, 2019. 2

[91] Xingxing Zuo, Xiaojia Xie, Yong Liu, and Guoquan Huang. Robust visual slam with point and line features. In *IROS*, 2017. 2