Supplementary Material for “Consistent 3D Line Mapping”

Xulong Bai[1,2,3], Hainan Cui[1,2,3*], and Shuhan Shen[1,2,3*]

[1] Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] CASIA-SenseTime Research Group
baixulong2022@ia.ac.cn, {hncui,shshen}@nlpr.ia.ac.cn

We provide supplementary material for the main paper. The content is presented according to the following list.

- In Sec. A, we present the experimental details and more comparisons with the state-of-the-art method L3D++ [5] and LIMAP [8].
- In Sec. B, we provide additional evaluations on the best proposal selection module, evaluate the equality of invalid and valid best proposals, visualize all best proposals, present evaluations on consistency percentage, and give the results by initializing our iterative algorithm with random best proposals.
- In Sec. C, we give the details on the line track building module and compare three build-in strategies of LIMAP [8] for generating line tracks.
- In Sec. D, we provide the details on joint optimization and visualize line maps with and without the imposition of coplanarity constraints.
- In Sec. E, we show an application of line map for line-assisted visual localization and perform performance comparisons using line maps generated by LIMAP [8] and our method, respectively.
- In Sec. F, we show an application of line map for SfM poses refinement, and perform performance comparisons using line maps generated by L3D++ [5], LIMAP [8] and our method, respectively.
- In Sec. G, we explain current limitations and future work.

A Experimental Details

A.1 Datasets

We consider both the synthetic dataset Hypersim [12] and the real dataset *Tanks and Temples* [7] for quantitative and qualitative experiments.

Hypersim [12] is a photorealistic synthetic dataset for holistic indoor scene understanding. The images of the first 8 scenes are used for evaluation and each image is resized to a maximum dimension of 800. We use COLMAP [14] to triangulate SfM points and build point tracks given Ground Truth (GT) camera poses. We use a visible GT point cloud for quantitative evaluation, which is generated by GT depth maps and GT poses using LIMAP library [8].

[*] Corresponding author

***Tanks and Temples* [7]** is a benchmark for image-based 3D reconstruction. Images in training data are used for evaluation, except for the scene *Ignatius* as it has almost no line structures. We use their GT point cloud for quantitative evaluation. Since the provided GT point cloud is focused on the main object, we follow the suggestion of LIMAP [8] and compute an axis-aligned bounding box of the GT points, extend it by 0.1 meters in all three dimensions, and only evaluate the lines inside this region.

A.2 Hyperparameters

Our hyperparameters are as follows: we use $n_v = 20$, $K = 10$ for line matching, $t_s = 1.5$, $t_c = 30\%$ for best proposal selection, $t_a = 2$ degrees, $t_p = 2$ pixels, and $t_o = 0$ for line track building. We default to using GlueStick line matcher [11] for searching top K line matches. It is also easy to substitute other customized line detectors and matchers.

A.3 Additional Comparisons with L3D++ [5] and LIMAP [8]

In the main paper, we use the open-source implementations of L3D++ [5] and LIMAP [8] for evaluation. We only evaluate the line tracks that have at least 4 support images, which is the same as the evaluation pipeline of LIMAP [8]. For the sake of fairness, when evaluating LIMAP [8], we use its four build-in methods (*i.e.*, Line+Line, Multiple Points, Line+Point, and Line+VP) for proposal generation, use its default *greedy* strategy for line track building, and jointly optimize points, lines, and vanishing points. We use the default line matcher of L3D++ [5], which is based on weak epipolar constraints. We use the default line matcher of LIMAP [8], which is GlueStick [11] and the same as our default. All methods use the same n_v and K for line matching.

Considering fairness, we additionally modify the code of L3D++ [5] to use the same top K line matches as our approach and LIMAP's, derived from GlueStick line matcher [11]. The corresponding results are detailed in Tabs. 1 and 2. Consistent with our primary observation in the main paper, within the Hypersim dataset [12], the length recall of our method is significantly higher than other methods, almost double theirs, albeit with a slightly lower inlier percentage compared to L3D++ [5]. In the *Tanks and Temples* dataset [7], our method significantly outperforms all competitors in terms of both scene completeness and accuracy, while also showcasing longer line tracks, demonstrating its superiority.

B Additional Evaluations on Best Proposal Selection

B.1 Additional Evaluation on Best Proposals

We evaluate all (*i.e.*, valid + invalid) best proposals and the valid best proposals selected by non-iterative method [8] or our iterative method on Hypersim dataset [12]. The corresponding results are reported in Tab. 3. Note that for

Table 1: Line reconstruction using the same detected LSD/DeepLSD lines and top K line matches generated from GlueStick line matcher on Hypersim [12]. The length recall at τ ($R\tau$), inlier percentage at τ ($P\tau$), and average supports are reported.

Line type	Method	R1	R5	R10	P1	P5	P10	# supports
LSD [16]	L3D++ [5]	30.2	168.9	224.1	**64.5**	**84.8**	**89.4**	17.5 / 20.9
	LIMAP [8]	36.2	209.1	278.2	62.1	82.3	87.4	16.8 / 19.7
	Ours	**58.2**	**303.1**	**388.0**	62.7	83.5	88.5	**19.9** / **21.8**
DeepLSD [9]	L3D++ [5]	38.8	193.8	252.6	**68.6**	**85.2**	**89.4**	16.3 / 19.2
	LIMAP [8]	32.8	180.6	237.6	64.9	81.5	86.0	18.7 / **26.2**
	Ours	**79.7**	**375.5**	**464.6**	64.8	83.3	88.1	**20.5** / 21.6

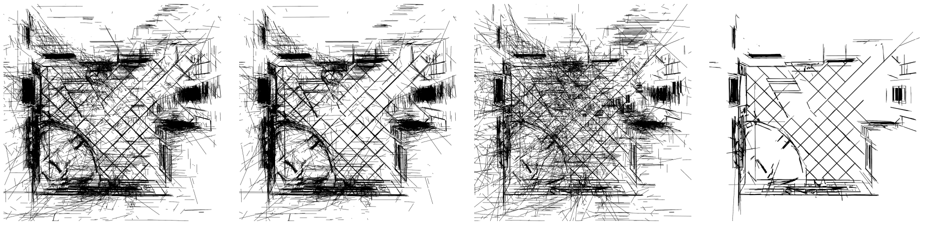
Table 2: Line reconstruction using the same detected LSD/DeepLSD lines and top K line matches generated from GlueStick line matcher on *train* split of *Tanks and Temples* [7]. The length recall at τ ($R\tau$), inlier percentage at τ ($P\tau$), and average supports are reported.

Line type	Method	R5	R10	R50	P5	P10	P50	# supports
LSD [16]	L3D++ [5]	460.2	1050.3	3612.5	43.6	56.8	86.4	9.2 / 10.2
	LIMAP [8]	593.2	1400.3	4990.1	44.1	57.0	**86.5**	8.9 / 10.0
	Ours	**935.0**	**2162.4**	**7147.2**	**45.4**	**58.3**	85.9	**13.1** / **13.6**
DeepLSD [9]	L3D++ [5]	425.4	983.0	3402.9	44.3	57.3	**86.2**	9.1 / 10.1
	LIMAP [8]	542.7	1259.9	4617.8	44.0	54.9	81.3	11.1 / 13.9
	Ours	**986.9**	**2255.7**	**7379.0**	**47.3**	**58.5**	83.7	**15.1** / **15.7**

Table 3: Best proposals evaluation on Hypersim [12] with DeepLSD [9] lines. The length recall at τ ($R\tau$), inlier percentage at τ ($P\tau$), and average supports are reported.

Best proposal type	Method	R1	R5	R10	P1	P5	P10	# supports
Valid + Invalid	Non-iterative [8]	879.7	4966.2	7023.7	46.8	63.8	70.7	1 / 1
	Iterative (Ours)	**1116.4**	**6258.6**	**8419.1**	**50.2**	**66.9**	**72.6**	1 / 1
Valid	Non-iterative [8]	878.8	4958.2	7006.7	48.9	66.6	73.6	1 / 1
	Iterative (Ours)	**1110.3**	**6200.3**	**8287.0**	**63.9**	**83.5**	**89.4**	1 / 1

the non-iterative method, there exist trivial supporting 2D line segments, which are matched 2D line segments generating the supported proposals. We ignore trivial supporting 2D line segments in all scenarios involving supporting 2D line segments. The "# supports" is "1 / 1" because each best proposal is outputted as a 3D line segment corresponding to a line track whose unique track element is the 2D line segment to which the best proposal belongs. As depicted in Tab. 3, after filtering out invalid best proposals, both the non-iterative method and our method achieve a significant enhancement in the inlier percentage, albeit at a slight sacrifice in length recall. This demonstrates the unreliability of invalid best



(a) Valid + invalid best proposals from non-iterative method [8]

(b) Valid best proposals from non-iterative method [8]

(c) Valid + invalid best proposals from our iterative method

(d) Valid best proposals from our iterative method

Fig. 1: Best proposals on the scene *ai_001_001* [12] with DeepLSD [9] lines. The length recall and inlier percentage at 1mm are as follows. (a): R1 = 847.7, P1 = 49.2. (b): R1 = 846.3, P1 = 53.3. (c): R1 = 1009.8, P1 = 49.5. (d): R1 = 1002.4, P1 = 69.3.

proposals. Furthermore, unlike the non-iterative method, our iterative approach substantially improves both length recall and inlier percentage, affirming the effectiveness of the proposed iterative algorithm in selecting more accurate best proposals.

B.2 Visualization of Best Proposals

Fig. 1 illustrates all best proposals and valid best proposals selected by non-iterative method [8] or our iterative method on the scene *ai_001_001* of Hypersim dataset [12], along with the length recall and inlier percentage at 1mm on the best proposals. Our method achieves the highest length recall and inlier percentage both on all best proposals and valid best proposals, and significantly reduces the noise on valid best proposals, again demonstrating its effectiveness.

B.3 Additional Evaluation on Consistency Percentage

We evaluate the average consistency percentage (refer to Section 4.2 in the main paper for its definition) on valid best proposals on Hypersim dataset [12]. The corresponding results are reported in Tab. 4. As illustrated in Tab. 4, our iterative method significantly improves the consistency percentage, demonstrating it can enhance the consistency of the support relationships.

Table 4: Average consistency percentage at 1°/5mm, 5°/10mm, and 10°/50mm on the valid best proposals on Hypersim [12] with DeepLSD [9] lines.

Method	CP1°/5mm	CP5°/10mm	CP10°/50mm
Non-iterative [8]	10.1	25.4	56.4
Iterative (Ours)	**30.7**	**64.3**	**96.9**

B.4 Initialization with Random Best Proposals

In the main paper, we commence our proposed iterative algorithm (Eq. (4) in the main paper) by initializing it with the best proposals selected through the non-iterative method [8] (Eq. (1) in the main paper). Additionally, in Tab. 5, we present the results of line reconstruction utilizing randomly selected best proposals for initialization. All results are presented without joint optimization. As illustrated in Tab. 5, the direct utilization of random best proposals for generating line tracks produces the worst length recall and average supports (refer to the 1^{st} row in Tab. 5). Nonetheless, the inlier percentage remains relatively high, because we filter out poor 3D line segments after generating line tracks. Following the execution of our iterative algorithm with the random best proposals as input, a significant enhancement is observed in length recall and average supports, albeit with a slight compromise in inlier percentage (refer to the 2^{nd} row in Tab. 5). This demonstrates the efficacy of our iterative algorithm in improving performance even when initialized with random best proposals. Importantly, it is noteworthy that despite the utilization of random best proposals for initialization, the attained length recall and average supports surpass those of L3D++ [5] and LIMAP [8] as depicted in Tab. 1 of the main paper. However, commencing our proposed iterative algorithm by initializing it with the best proposals obtained from the non-iterative method [8] achieves higher length recall and inlier percentage compared with random initialization, demonstrating the sensitivity of our iterative algorithm to initialization.

Table 5: Line reconstruction using different initial best proposals for our iterative algorithm on Hypersim [12] with DeepLSD [9] lines. The length recall at τ ($R\tau$), inlier percentage at τ ($P\tau$), and average supports are reported.

Initial best proposals type	Method	R1	R5	R10	P1	P5	P10	# supports
Random	w/o iteration	17.5	87.1	114.3	**69.7**	**85.6**	**89.8**	9.1 / 9.3
Random	w/ iteration	43.1	224.1	299.4	65.1	80.9	85.7	**22.1** / **23.5**
From non-iterative method [8]	w/o iteration	54.44	303.3	414.7	61.9	80.1	85.9	16.9 / 17.7
From non-iterative method [8]	w/ iteration	**63.3**	**340.3**	**452.7**	65.8	82.9	87.7	20.5 / 21.6

C Details on Line Track Building

Given the best proposal of all 2D line segments, we first build line tracks by clustering 2D line segments according to their best proposals. Secondly, we estimate the new 3D line segment for each line track. Lastly, we execute track remerging and filtering to refine line tracks.

C.1 3D Line Segment Estimation of Line Track

Given a line track and its corresponding best proposals, we apply the Principal Component Analysis (PCA) method for all directions of the best proposals and use the principal eigenvector and mean 3D endpoints of the best proposals to fit a 3D infinite line. Then we project the 3D endpoints of all these best proposals onto the fitted 3D infinite line and take the third outermost endpoints on both sides as the two new 3D endpoints. After joint optimization, we use the same method to get the new 3D line segment from the optimized 3D infinite line.

C.2 Track Remerging and Filtering

Given all line tracks, we follow [8] to iteratively remerge two line tracks by a stricter similarity threshold and re-estimate the new line track. The 2D collinearity constraints of line track elements are also used in this step to avoid two non-collinear 2D line segments belonging to the same new line track. Lastly, we also filter line tracks based on reprojection error.

C.3 Additional Comparisons with LIMAP [8]

In the main paper, for LIMAP [8], we use its default *greedy* strategy to generate line tracks. The *greedy* strategy is the version of removing our 2D collinearity constraints. Besides the *greedy* strategy, LIMAP [8] also provides two other built-in strategies (*exhaustive* and *avg*) to merge the best proposals for generating line tracks. The *exhaustive* strategy tries to avoid merging two groups if there are two best proposals from different groups that exhibit overlap but lack collinearity. The *avg* strategy tries to avoid merging two groups if the respective average 3D line segments of the two groups are not collinear. Although all of *exhaustive*, *avg*, and our strategies try to impose constraints for generating line tracks, an important difference is that our 2D collinearity constraints are directly established by image observations, while the constraints of *exhaustive* and *avg* strategies are established in 3D space. Therefore, our 2D collinearity constraints are more consistent with image observations.

We substitute our strategy with the *greedy*, *exhaustive*, or *avg* strategy of LIMAP [8] in our line track building module. The corresponding evaluations are reported in Tab. 6. All experiments are without joint optimization. Our method is better on length recall at all thresholds, while our inlier percentage is slightly lower than the *greedy* method at 1mm and 5mm. As discussed in the main paper, our method decreases the average number of supports across all line tracks because the inconsistent track elements were assigned to different consistent line tracks.

D Details on Joint Optimization

D.1 Parameterization

In optimization, we optimize 3D infinite lines formed by each initial 3D line segment. For 3D infinite line, Plücker coordinates [8] $(\boldsymbol{d}, \boldsymbol{m})$ is used for com-

Table 6: The results of using different line track building methods on Hypersim [12] with DeepLSD [9] lines. The length recall at τ ($R\tau$), inlier percentage at τ ($P\tau$), and average supports are reported.

Method	R1	R5	R10	P1	P5	P10	# supports
Greedy [8]	36.5	188.4	242.2	**68.3**	**83.3**	86.9	**22.3** / **33.2**
Exhaustive [8]	51.7	275.8	363.8	64.4	81.9	87.1	18.5 / 21.9
Avg [8]	38.7	203.2	262.0	67.6	83.1	87.1	21.8 / 32.1
Collinearity2D (Ours)	**63.3**	**340.3**	**452.7**	65.8	82.9	**87.7**	20.5 / 21.6

putation where $\boldsymbol{d}$ is the normalized direction of the 3D line and $\boldsymbol{m} = \boldsymbol{p} \times \boldsymbol{d}$ is the moment ($\boldsymbol{p}$ is any point along the line), and orthonormal representation [8] $(\boldsymbol{U}, \boldsymbol{W}) \in SO(3) \times SO(2)$ (4 DoF) is used in optimization to avoid over-parameterization. For 3D infinite plane, Hesse form $(\boldsymbol{n}_\pi, d_\pi)$ is used for computation where $\boldsymbol{n}_\pi$ is the normalized normal vector of the plane and $d_\pi = \boldsymbol{n}_\pi \cdot \boldsymbol{p}$ is the shortest distance from the origin to plane ($\boldsymbol{p}$ is any point on the plane), and closest point plane representation [4] (3 DoF) is used in optimization to avoid over-parameterization. For 3D vanishing points, we parameterize them with a 3-dimensional homogeneous vector (2 DoF). Therefore, the variables of the final problem exhibit $3N_P + 4N_L + 3N_\Pi + 2N_{VP}$ degrees of freedom in total.

D.2 Details on All Energy Terms

For E_P, E_L, E_{PL} and E_{LVP}, we use the same formulation as LIMAP [8]. Our contribution lies in introducing $E_{P\Pi}$ and $E_{L\Pi}$, which encodes the coplanarity constraints overlooked by classical 3D line mapping methods [5, 6, 8].

Data Term for the Point Tracks (E_P). This term is the sum of the squared reprojection error for all point tracks.

Data Term for the Line Tracks (E_L). This term is the sum of the squared reprojection error for all line tracks, which can be written as follows:

$$E_L = \sum_{L \in \mathcal{L}} \sum_k w_\angle^2(L_k, l_k) \cdot e_{\text{perp}}^2(L_k, l_k), \tag{1}$$

$$w_\angle(L_k, l_k) = \exp\left(\alpha(1 - \cos\left(\angle(L_k, l_k)\right))\right), \tag{2}$$

where $\mathcal{L}$ denotes all 3D infinite lines formed by the corresponding initial 3D line segments, l_k is the detected 2D line segment (*i.e.*, k^{th} line track element), L_k is the projected 2D infinite line of the 3D infinite line L, $e_{\text{perp}}(L_k, l_k)$ computes the root of the sum of the squared perpendicular distance of two endpoints of l_k to L_k. α equals 10.0 in our system.

3D Point-Line Association Term (E_{PL}). This term is the sum of the squared distance between a 3D point and the associated 3D infinite line, weighted by the number of 2D connections among supports of the corresponding 3D point-line association.

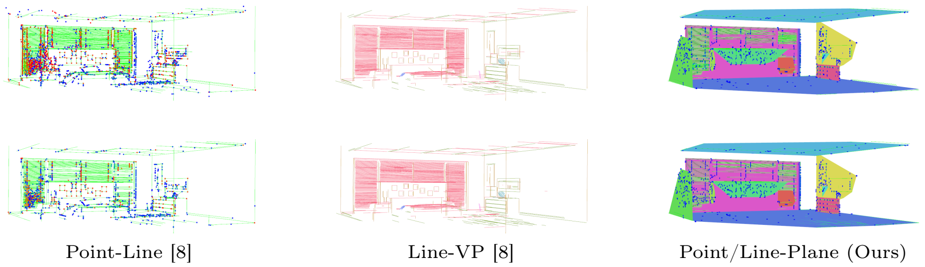
3D Point-Plane Association Term ($E_{P\Pi}$). This term is the sum of the squared distance between a 3D point and the associated 3D plane, weighted by the number of image supports of the corresponding point track.
3D Line-Plane Association Term ($E_{L\Pi}$). This term is the sum of the squared distance between a 3D point on the 3D infinite line and the associated 3D plane, and the squared cosine of the angle between the direction of the 3D infinite line and the normal vector of the associated 3D plane, weighted by the number of image supports of the corresponding line track. As discussed in the main paper, we use our proposed coordinate-independent method to determine the 3D point on the 3D infinite line during optimization.
3D Line-VP Association and VP Orthogonality Regularization Term (E_{LVP}). A part of this term is the squared sine of the direction angle between the 3D infinite line and the associated 3D vanishing point, weighted by the number of 2D connections among supports for each 3D line-VP association. Another part of this term is the sum of the squared cosine of the angle between two 3D nearly orthogonal vanishing points.

All soft associations are optimized with robust Huber loss using Ceres [1].

D.3 Details on Establishing 3D Association

We follow the method of LIMAP [8] for establishing initial 3D point-line associations, line-VP associations, and VP orthogonality regularization. However, due to the incorporation of coplanarity constraints, we must explore an appropriate method for establishing initial point-plane and line-plane associations.
3D Point-Line Association. To build the initial 3D point-line association, we identify the pair of the point track and the line track that has at least 3 connected edges on the 2D point-line association graphs among their 2D supports.
3D Point-Plane Association. We first detect 3D planes from the point cloud [2], which consists of SfM points and sampled points extracted from the initial line map. Then we utilize the inlier points of each 3D plane to build the initial 3D point-plane association.
3D Line-Plane Association. Same as the above method, we utilize the inlier points of each plane to build the initial 3D line-plane association. A difference is that the sampled points of a 3D line segment may be associated with different planes. In this case, we establish associations between the line and each plane, reflecting the reality that a single line can intersect with multiple planes.
3D Line-VP Association and VP Orthogonality Regularization. To build the initial line-VP association, we identify the pair of the VP track and the line track that has at least 3 connected edges on the 2D line-VP association graphs among their 2D supports. The VP track is built by line tracks and 2D line-VP associations. For the initial VP orthogonality regularization, we collect pairs of VPs that are nearly orthogonal (angles exceeding 87 degrees).

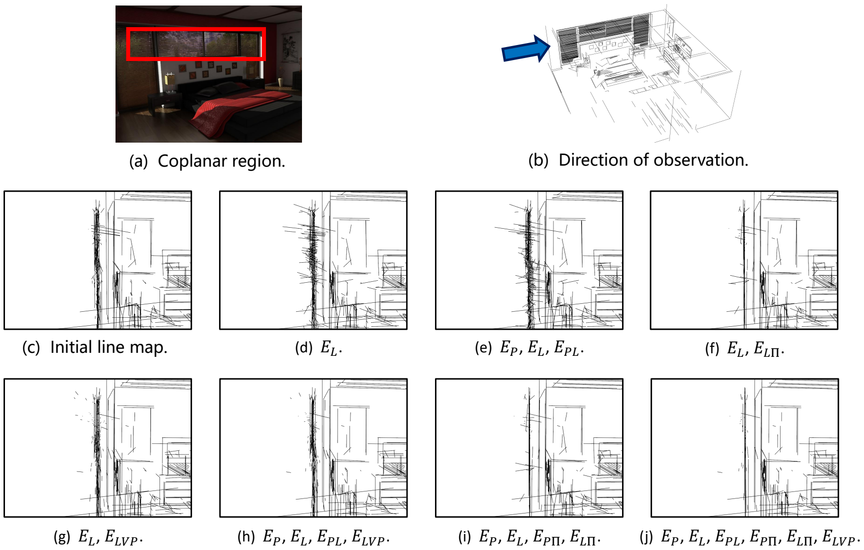
For every 3D association, after optimization, we can optionally output the final 3D association by removing the outlier 3D association based on the thresholds, similar to LIMAP [8]. Some examples of initial and final associations for the scene *ai_001_006* on the Hypersim dataset [12] are illustrated in Fig. 2.


Point-Line [8]

Line-VP [8]

Point/Line-Plane (Ours)

Fig. 2: 3D associations for the scene *ai_001_006* on the Hypersim dataset [12]. In these examples, we only show geometric entities that have association. *Top row*: Initial 3D associations. *Bottom row*: Optional final 3D associations that have removed outliers based on the optimized geometric entities and thresholds. The points associated with one line or more than one are colored in blue or red respectively in point-line association, while parallel lines associated with the same VP are colored the same in line-VP association. Each plane has a different color in point-plane and line-plane association.


(a) Coplanar region.

(b) Direction of observation.

(c) Initial line map.

(d) E_L.

(e) E_P, E_L, E_{PL}.

(f) E_L, $E_{L\Pi}$.

(g) E_L, E_{LVP}.

(h) E_P, E_L, E_{PL}, E_{LVP}.

(i) E_P, E_L, $E_{P\Pi}$, $E_{L\Pi}$.

(j) E_P, E_L, E_{PL}, $E_{P\Pi}$, $E_{L\Pi}$, E_{LVP}.

Fig. 3: Visualization of coplanarity constraint effects. (a): The approximate coplanar region in the scene *ai_001_006* on the Hypersim dataset [12]. (b): The direction of observation. (c)-(j): Side views of reconstructing the approximate coplanar region by minimizing distinct combinations of energy terms in the process of joint optimization.

D.4 Visualization of Coplanarity Constraint Effects

As discussed in the main paper, we introduce coplanarity constraints (*i.e.*, $E_{P\Pi}$ and $E_{L\Pi}$) to 3D line mapping, which is demonstrated to be crucial for line reconstruction in our ablation study (refer to Tab. 6 in the main paper). In Fig. 3, we additionally present an example of optimizing 3D line segments based on various energy terms, aiming to better represent the approximate coplanar region. We present eight results delineating the approximate coplanar region in the scene *ai_001_006* on the Hypersim dataset [12], observed from a lateral perspective (as shown in Fig. 3 (b)). Each result minimizes a distinct combination of energy terms, which is consistent with the order of the ablation study in Tab. 6 in the main paper. Note that we do not filter any 3D line segment in this example. The initial 3D point-line, point-plane, line-plane, and line-VP associations for joint optimization are depicted in the top row of Fig. 2.

As illustrated in Fig. 3, it is difficult to recover coplanar regions without the help of coplanar constraints (as shown in Fig. 3 (c), (d), (e), (g) and (h)). When integrating coplanar constraints, the reconstructed 3D line segments were significantly coplanar (as shown in Fig. 3 (f), (i) and (j)). In the main paper, we jointly optimize points, lines, planes, and vanishing points, resulting in visually superior outcomes in this example (as shown in Fig. 3 (j)).

E Application: Line-Assisted Visual Localization

We utilize point maps obtained from HLoc [13] and line maps obtained from our proposed method to conduct line-assisted visual localization through the localization pipeline presented in LIMAP [8]. We compared the performance of executing line-assisted visual localization tasks using line maps obtained from our method to those obtained from LIMAP's approach [8]. We utilize LSD line detector [16] and SOLD2 line matcher [10] to reconstruct line maps, which is the default configuration for the localization task on 7Scenes dataset [15] in LIMAP [8]. Additionally, we compare the performance of point-based visual localization obtained from HLoc [13]. The corresponding results are reported in Tab. 7. Our method consistently outperformed the results of using line maps obtained from LIMAP [8], demonstrating the effectiveness of our method.

Table 7: Per-scene results of visual localization on 7Scenes [15]. The median translation and rotation error in cm and degrees are represented by e_t and e_r, respectively. We also report the pose accuracy (%) at a 5cm / 5deg threshold.

Scene	# Query images	HLoc [13]	LIMAP [8]	Ours
		e_t / e_r / acc.(%)	e_t / e_r / acc.(%)	e_t / e_r / acc.(%)
Chess	2000	**2.4** / **0.84** / **93.0**	2.5 / 0.85 / 92.3	2.5 / 0.85 / 92.8
Fire	2000	2.3 / 0.89 / 88.9	2.1 / **0.84** / **95.5**	**2.0** / **0.84** / **95.5**
Heads	1000	1.1 / 0.75 / **95.9**	1.1 / 0.76 / **95.9**	**1.0** / **0.73** / 95.3
Office	4000	3.1 / 0.91 / 77.0	**3.0** / 0.89 / 78.4	**3.0** / **0.88** / **79.5**
Pumpkin	2000	5.0 / 1.32 / 50.4	4.7 / 1.23 / 52.9	**4.6** / **1.20** / **53.2**
Redkitchen	5000	4.2 / **1.39** / 58.9	**4.1** / **1.39** / 60.2	**4.1** / **1.39** / **60.7**
Stairs	1000	5.2 / 1.46 / 46.8	3.7 / 1.02 / 71.1	**3.6** / **0.97** / **72.8**
Avg.	2429	3.3 / 1.08 / 73.0	**3.0** / 1.00 / 78.0	**3.0** / **0.98** / **78.5**

F Application: Refining Structure-from-Motion

Furthermore, we display another application of line maps for refining the camera poses obtained from the point-based structure-from-motion method, *e.g.*, COLMAP [14]. We first run COLMAP [14] with SuperPoint [3] on Hypersim dataset [12] to get camera intrinsics and poses. Then we reconstruct 3D line maps to obtain 3D line segments and their track information using L3D++ [5], LIMAP [8] and our method, respectively, where each line mapping method receives the same 2D line segments and top K line matches as input. Lastly, we use the obtained 3D line segments and their track information to perform joint bundle adjustment of points and lines to refine intrinsics and poses.

We evaluate the absolute pose errors compared with Ground Truth poses. The original pose errors of COLMAP [14], along with the refined pose errors, are reported in Tab. 8. As depicted in Tab. 8, the joint bundle adjustment of points and lines markedly improves original pose accuracy [14]. Unlike alternative line mapping approaches [5, 8], employing our line maps for joint bundle adjustment achieves the least pose errors, confirming the capability of our line mapping method to generate more accurate 3D line segments and robust tracks.

Table 8: Per-scene results of SfM poses refinement on Hypersim [12]. The median translation and rotation error in mm and degrees are represented by e_t and e_r, respectively. We also report the pose accuracy (%) at a 5mm / 0.5deg threshold.

Scene	COLMAP [14]	[14] + L3D++ [5]	[14] + LIMAP [8]	[14] + Ours
	e_t / e_r / acc.(%)	e_t / e_r / acc.(%)	e_t / e_r / acc.(%)	e_t / e_r / acc.(%)
ai_001_001	3.1 / 0.078 / 91.8	2.4 / 0.053 / 96.9	2.1 / 0.052 / **97.9**	**1.9** / **0.043** / **97.9**
ai_001_002	2.8 / 0.079 / 67.7	3.2 / 0.100 / 73.7	2.0 / 0.043 / 86.9	**1.7** / **0.040** / **91.9**
ai_001_003	5.3 / 0.121 / 47.0	3.9 / 0.088 / 69.0	3.7 / 0.094 / 75.0	**3.4** / **0.069** / **79.0**
ai_001_004	3.2 / 0.063 / 76.0	2.2 / 0.048 / 87.0	2.1 / **0.044** / 88.0	**1.9** / 0.047 / **90.0**
ai_001_005	3.3 / 0.072 / 77.0	3.2 / 0.067 / 79.0	3.1 / 0.058 / 74.0	**2.1** / **0.042** / **90.0**
ai_001_006	2.9 / 0.052 / 74.0	2.5 / 0.041 / 84.0	2.2 / **0.037** / 85.0	**1.9** / 0.052 / **89.0**
ai_001_007	2.0 / 0.071 / **81.0**	1.8 / 0.062 / **81.0**	1.9 / 0.058 / **81.0**	**1.7** / **0.049** / **81.0**
ai_001_008	6.7 / 0.164 / 36.0	5.8 / 0.156 / 46.0	4.9 / 0.098 / 51.0	**3.5** / **0.064** / **74.0**
Avg.	3.7 / 0.088 / 68.8	3.1 / 0.077 / 77.1	2.8 / 0.061 / 79.9	**2.3** / **0.051** / **86.6**

G Limitations and Future Work

Our proposed iterative algorithm, aimed at selecting the best proposals, relies on the initial best proposals. As outlined in Sec. B.4, unlike other methods [5, 8], our approach achieves superior length recall and average support counts across all line tracks, even when employing a randomly selected best proposal as the initial value. However, optimal performance is achieved when utilizing the best proposals obtained through the non-iterative method [8] as the initial value. In the future, our objective is to develop a more robust strategy for acquiring the initial best proposals.

Furthermore, within the joint optimization module, we perform the fitting of 3D planes directly from the point cloud, without incorporating image observations. In the future, we aim to integrate image observations into the process of fitting 3D planes. This enhancement will enable us to more effectively rectify errors that may have accumulated during the reconstruction process.

References

1. Agarwal, S., Mierle, K., Team, T.C.S.: Ceres Solver (10 2023)
2. Araújo, A.M., Oliveira, M.M.: A robust statistics approach for plane detection in unorganized point clouds. Pattern Recognition **100**, 107115 (2020)
3. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 224–236 (2018)
4. Geneva, P., Eckenhoff, K., Yang, Y., Huang, G.: Lips: Lidar-inertial 3d plane slam. In: IEEE International Conference on Intelligent Robots and Systems (IROS). pp. 123–130 (2018)
5. Hofer, M., Maurer, M., Bischof, H.: Efficient 3d scene abstraction using line segments. Computer Vision and Image Understanding (CVIU) **157**, 167–178 (2017)
6. Jain, A., Kurz, C., Thormählen, T., Seidel, H.P.: Exploiting global connectivity constraints for reconstruction of 3d line segments from images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1586–1593 (2010)
7. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)
8. Liu, S., Yu, Y., Pautrat, R., Pollefeys, M., Larsson, V.: 3d line mapping revisited. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21445–21455 (2023)
9. Pautrat, R., Barath, D., Larsson, V., Oswald, M.R., Pollefeys, M.: Deeplsd: Line segment detection and refinement with deep image gradients. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17327–17336 (2023)
10. Pautrat, R., Lin, J.T., Larsson, V., Oswald, M.R., Pollefeys, M.: Sold2: Self-supervised occlusion-aware line description and detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11368–11378 (2021)
11. Pautrat, R., Suárez, I., Yu, Y., Pollefeys, M., Larsson, V.: Gluestick: Robust image matching by sticking points and lines together. In: IEEE International Conference on Computer Vision (ICCV). pp. 9706–9716 (2023)
12. Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: IEEE International Conference on Computer Vision (ICCV). pp. 10912–10922 (2021)
13. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12716–12725 (2019)
14. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4104–4113 (2016)

15. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2930–2937 (2013)
16. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **32**(4), 722–732 (2008)