IncreLM: Incremental 3D Line Mapping

Xulong Bai[1,2,3]
baixulong2022@ia.ac.cn
Hainan Cui*[1,2,3]
hncui@nlpr.ia.ac.cn
Shuhan Shen*[1,2,3]
shshen@nlpr.ia.ac.cn

[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] CASIA-SenseTime Research Group, Beijing, China

Abstract

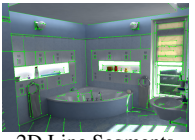
Given posed images and Structure-from-Motion (SfM) points, we aim to produce 3D line segments (3D LSs) and line tracks. Traditional methods typically reconstruct a single 3D LS corresponding to each 2D line segment (2D LS) independently and in parallel, later merging all single 3D LSs to produce final 3D LSs and line tracks. However, this independence may lead to inconsistencies in outlier line matches identified across different single 3D LS reconstructions. To enhance the robustness of outliers, we propose an incremental 3D line mapping method that sequentially reconstructs each final 3D LS, using outlier line matches identified from earlier reconstructions to guide later ones. In our approach, 3D LS hypotheses are generated through two-view line triangulation, utilizing 3D points and vanishing points within a hybrid RANSAC framework. A graph is then created, with nodes representing the hypotheses and edges linking nodes that share identical source 2D LSs. Initially, the best 3D LS hypothesis is found based on neighborhood supports and added to an empty 3D line map. We then extend the line track of the new 3D LS and filter out outlier matches through reprojection. Next, we filter out outlier nodes generated by outlier matches, locate the next-best 3D LS hypothesis, and integrate it into the existing map. This iterative process continues until no best 3D LS hypotheses can be identified from the graph. Finally, the line tracks are merged and a joint optimization is performed to improve the map quality. Experiments show that our system exceeds current state-of-the-art methods in completeness and accuracy and produces longer line tracks. Code is available at https://github.com/3dv-casia/IncreLM.

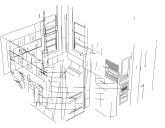
1 Introduction

Generally, most 3D reconstruction algorithms [2, 20, 28] use 3D points to represent the scene structure. However, using 3D points to represent weakly textured areas and slender structures is difficult and redundant. Hence, the 3D line segments (3D LSs) become popular recently as they can effectively and concisely represent these scenes in man-made environments.

With the known camera poses, traditional 3D line mapping methods, *e.g.*, L3D++ [11] and LIMAP [19], reconstruct the single 3D LS corresponding to each 2D line segment (2D

*Corresponding author


2D Line Segments


L3D++ [11]


LIMAP [19]

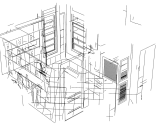

IncreLM

Figure 1: This figure shows the 3D line mapping results on a sample Hypersim dataset [26].

LS) independently and in parallel, later merging all single 3D LSs to produce final 3D LSs and line tracks. In this paper, a *line track* is defined as a set of 2D LSs in images, corresponding to certain parts of the same 3D LS entity. However, this independence may lead to inconsistencies in outlier line matches identified across different single 3D LS reconstructions, making current methods incapable of finding real outliers. To improve the scene completeness, traditional methods usually take one-to-many line matches as inputs, where one 2D LS can match multiple candidate 2D LSs on a neighboring image. As a result, the input line matches are contaminated by plenty of outliers. In traditional methods [11, 19], handling numerous outlier matches in parallel is challenging and restricts the system's robustness.

Inspired by the incremental SfM method [28], known for producing accurate and robust reconstruction results, we propose a novel incremental 3D line mapping framework. To improve the scene completeness, our framework also takes one-to-many line matches as inputs. In contrast to conventional frameworks, our framework sequentially reconstructs each final 3D LS, using outlier line matches identified from earlier reconstructions to guide later ones. To implement this incremental framework, we first generate 3D LS hypotheses through two-view line triangulation on matches. However, traditional line triangulation is unstable when 3D LS is near the epipolar plane. To address this degeneracy issue, we propose a robust two-view line triangulation method leveraging 3D points and vanishing points (VPs) within a hybrid RANSAC framework [3]. A graph is then created, with nodes representing the hypotheses and edges linking nodes that share identical source 2D LSs. Utilizing this graph, we develop a strength score function for nodes, which encodes geometric supports from neighboring nodes. Initially, the best 3D LS hypothesis is found with the largest strength score and added to an empty 3D line map. We then extend the line track of the newly added 3D LS using line matches and filter out outlier matches through reprojection. Next, we filter out outlier nodes generated by outlier matches, locate the next-best 3D LS hypothesis, and integrate it into the existing map. This iterative process continues until no best 3D LS hypotheses can be identified from the graph. During the incremental mapping, the outlier matches are gradually detected by 3D LS outputs and ignored in the further mapping process. This feedback mechanism enhances the system's robustness to outliers. Finally, we propose a line track merging module that creates longer line tracks, thereby facilitating the construction of a more precise map through joint optimization.

In conclusion, our **Contributions** are as follows: 1) An incremental 3D line mapping system, which takes one-to-many 2D LS correspondences as inputs and utilizes a feedback verification mechanism based on 3D LS outputs to iteratively handle outlier matches; 2) A robust two-view line triangulation algorithm, which leverages 3D points and vanishing points through a hybrid RANSAC framework; 3) A 3D LS hypotheses graph, which helps to find the order of incremental line mapping and is dynamically updated through the mapping process; 4) A line track merging module, which produces longer line tracks and contributes accurate optimization. Fig. 1 shows a sample comparison of the 3D LS mapping results produced by the state-of-the-art methods. Through this comparison, our method, abbreviated as IncreLM, reconstructs a more complete 3D LS map.

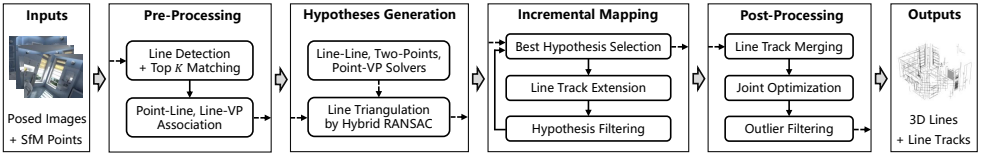


Figure 2: Pipeline. Our system begins with feature pre-processing and 3D LS hypotheses generation. It then incrementally produces 3D LSs and line tracks, one at a time. Finally, post-processing is conducted to enhance the overall quality of the mapping.

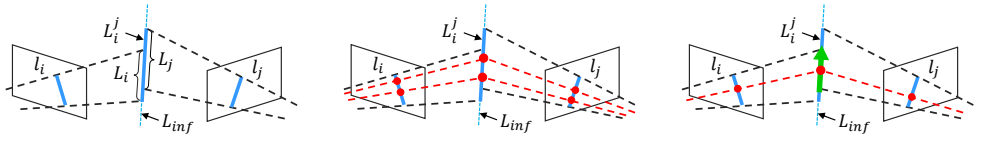
2 Related Work

3D Line Mapping Without Known Camera Poses. As a pioneering work, Weng *et al*. [34] presented the closed-form solution of SfM from line correspondences. Quan and Kanade [25] presented a method for affine SfM from line correspondences. Bartoli and Sturm [1] presented representations, triangulation, and bundle adjustment for a line-based SfM. Salaün *et al*. [27] estimated the relative scale of two successive bifocal calibrations. Micusik and Wildenauer [21] decoupled rotation and translation estimation. Zhang and Koch [38] introduced the Cayley representation of spatial lines in a line-based SfM. In addition to SfM, there are many simultaneous localization and mapping (SLAM) methods [4, 7, 12, 16, 17, 30, 32, 35, 36, 37, 39] use line features to improve localization accuracy. The target of the above methods mainly focuses on accurate calibration and localization. However, our work aims at producing a complete and accurate 3D LS map, especially in the case of existing too many outlier matches, so we focus on 3D line mapping with known camera poses.

3D Line Mapping With Known Camera Poses. Given camera poses, Jain *et al*. [13] presented a 3D LS reconstruction method without explicit line matches. Hofer *et al*. [8, 9, 10] extended Jain's work by introducing weak epipolar constraints and finally introduced the Line3D++ package [11], referred to as L3D++ in this paper. Wei *et al*. [33] presented a line matching method based on coplanar constraints, and the line mapping is performed by progressively selecting the representative 3D LSs. However, it does not produce line tracks. Guo *et al*. [5] presented a greedy algorithm for line mapping based on the line matching score. Liu *et al*. [19] introduced the library LIMAP, which employed 3D points and vanishing points for two-view line triangulation to tackle degenerate issues and conducted joint optimization on 3D points, 3D LSs, and VPs. Nevertheless, traditional methods [11, 19] typically handle each 3D LS corresponding to a 2D LS independently, the outliers detected between two 3D LSs may be inconsistent, making them not robust to outlier matches. In this paper, we propose a robust 3D line mapping system to produce 3D LSs one by one and iteratively filter outlier matches during the incremental process.

3 Incremental 3D Line Mapping

Given posed images and SfM points, we first detect 2D LSs and identify their top K matches in each neighboring image determined by co-visible SfM points. Each image has n_v neighboring images, for which we set $K = 10$ and $n_v = 20$. Then, we associate 2D feature points with 2D LSs and detect VPs. After that, we generate 3D LS hypotheses and perform incremental 3D line mapping to produce 3D LSs and line tracks one by one. Finally, we merge line tracks and perform joint optimization of 3D points, 3D LSs, and VPs. We also filter out outlier tracks to finalize the 3D LSs and line tracks. Our pipeline is illustrated in Fig. 2.



(a) Line-Line (b) Two-Points (c) Point-VP

Figure 3: Two-view line triangulation solvers. Each solver is used to estimate a 3D infinite line L_{inf}. (l_i, l_j) is a 2D LS correspondence. The red 3D points are associated SfM points. The red 2D points are observations of the SfM points. The green arrow is the 3D direction of the associated VP of one of the 2D LSs.

3.1 3D Line Segment Hypotheses Generation

Let L_i^j be the 3D LS hypothesis triangulated from a 2D LS correspondence (l_i, l_j), where l_i and l_j are two *source* 2D LSs. We term it a *hypothesis* since the 2D LS correspondence may be incorrect. In practice, as we utilize one-to-many matches, numerous outlier matches occur, leading to a significant number of outlier 3D LS hypotheses.

The traditional two-view line triangulation method, as utilized in LIMAP [19], employs four solvers (*i.e.*, Line + Line, Multiple Points, Line + Point, and Line + VP) to generate *multiple* 3D LS hypotheses from a 2D LS correspondence. This method then incorporates all generated 3D LS hypotheses into the reconstruction, which introduces redundancy and a high incidence of outliers. To address this challenge, we aim to triangulate *one* single, precise 3D LS hypothesis from a 2D LS correspondence. Additionally, to resolve the degeneracy issue (when the 3D line is near the epipolar plane [6]) and enhance triangulation accuracy, we utilize a hybrid RANSAC framework [3]. This approach integrates multiple solvers, effectively leveraging the associated SfM points and VPs of 2D LSs.

Two-view Line Triangulation. In our specific hybrid RANSAC, the *model* is defined as the 3D infinite line L_{inf}, and we have three kinds of *data*: 2D LS correspondence (l_i, l_j), 3D points set $\mathcal{P}$ which contains the associated SfM points of the 2D LSs, and vanishing points set $\mathcal{VP}$ which contains the associated VPs of 2D LSs. There are three *solvers*: the Line-Line solver [6], along with two solvers we propose: Two-Points and Point-VP. For the Line-Line solver, we take the intersection of the back-projected plane of l_i and l_j as the 3D infinite line L_{inf}, as illustrated in Fig. 3a. This solver is similar to the Line + Line solver presented in LIMAP [19]. For the Two-Points solver, we employ two 3D points to determine L_{inf}, as illustrated in Fig. 3b. For the Point-VP solver, we employ a 3D point and a 3D direction of an associated VP to determine L_{inf}, as illustrated in Fig. 3c. The classical Line-Line solver is intrinsically unstable in the degenerate case, while the two proposed solvers are not affected. In contrast to the Multiple Points, Line + Point, and Line + VP solvers presented in LIMAP [19], our proposed Two-Points and Point-VP solvers employ minimal data to estimate a 3D infinite line, which is more compatible with the RANSAC scheme.

Given the estimated model L_{inf}, we first verify whether the 2D LS correspondence (l_i, l_j) is an inlier. To this end, the endpoints of l_i and l_j are projected to L_{inf} to produce L_i and L_j, respectively. If L_i and L_j have no overlap (*i.e.*, l_i and l_j do not share a common view area) or if L_{inf} fails the reprojection test for l_i and l_j, then (l_i, l_j) is deemed an outlier and L_{inf} is discarded. To identify the inliers in $\mathcal{P}$, we first compute the perpendicular distance t from a 3D point $p \in \mathcal{P}$ to L_{inf}. Then, to obtain scale invariance, we compute a scale factor $\sigma = \text{median}(\sigma_k)$, where $\sigma_k = d_k / f_k$, and d_k is the depth of p in p's observed image and f_k is the focal length of this observed image, respectively. This scale factor encodes how far the

3D point can move in 3D before reaching 1-pixel error in the image. If the perpendicular distance t is lower than σ, the 3D point p is considered as an inlier. The inliers in $\mathcal{VP}$ are those vanishing points that form an angle of less than 5 degrees with L_{inf}.

The Line-Line solver produces a unique model because there is only one 2D LS correspondence data. Therefore, we initialize the hybrid RANSAC with the model estimated by the Line-Line solver and will not use the Line-Line solver again. In each RANSAC iteration, following the standard hybrid RANSAC framework [3], we estimate the model by selecting one of the Two-Points and Point-VP solvers that has not been used often previously. Then, the model is verified by l_i and l_j and scored with the inlier count overall on $\mathcal{P}$ and $\mathcal{VP}$. The best model will be updated if the new inlier count surpasses the number of inliers corresponding to the best model in the last iteration. Finally, the RANSAC stops when the Two-Points solver is chosen K_1 times or the Point-VP solver is chosen K_2 times, where

$$K_1 = \frac{\log(1-P)}{\log(1-\varepsilon_p^2)}, \qquad K_2 = \frac{\log(1-P)}{\log(1-\varepsilon_p\varepsilon_{vp})}, \tag{1}$$

the ε_p and ε_{vp} are the estimated inlier ratio in $\mathcal{P}$ and $\mathcal{VP}$, respectively. The probability P is set to 0.99. The final 3D LS hypothesis L_i^j is the union of the L_i and L_j of the best L_{inf} model. Consequently, after executing the Hypotheses Generation module, each 2D LS correspondence will have only one precise 3D LS hypothesis if any.

3.2 Incremental Mapping

If (l_i, l_j) is an inlier match and the two-view line triangulation is accurate, L_i^j will be an inlier hypothesis. To evalute L_i^j, we collect hypotheses $\{L_i^{m_i}\}$ triangulated from l_i and l_i's all matched 2D LSs $\{l_{m_i}\}$, as well as hypotheses $\{L_j^{m_j}\}$ triangulated from l_j and l_j's all matched 2D LSs $\{l_{m_j}\}$, to establish the set $\mathcal{H} = \{L_i^{m_i}\} \bigcup \{L_j^{m_j}\} \setminus \{L_i^j, L_j^i\}$. If L_i^j is an inlier hypothesis, there should be many other inlier hypotheses in $\mathcal{H}$ in the spatial vicinity of L_i^j, because all inlier hypotheses whose source 2D LS is l_i or l_j should be located in a similar 3D position. If L_i^j is an outlier hypothesis (usually because (l_i, l_j) is an outlier match), it is unlikely that there exist hypotheses in $\mathcal{H}$ close to L_i^j. Intuitively, the more hypotheses in $\mathcal{H}$ are close to L_i^j, the more geometric supports L_i^j receive, and the more likely L_i^j is to be an inlier hypothesis. However, outlier hypotheses in $\mathcal{H}$ may provide incorrect geometric support, degenerating the evaluation for L_i^j. To address this issue, we develop an incremental framework to iteratively filter out outlier hypotheses and extract the current best hypothesis as the 3D LS output.

Specifically, we introduce an undirected weighted graph $G = (\mathcal{V}, \mathcal{E})$ to encode all hypotheses and connections. The nodes $\mathcal{V}$ denote hypotheses and the initial edges $\mathcal{E}$ connect two nodes that share a common source 2D LS. Let $\mathcal{C}$ denote the set of nodes connected to L_i^j. In each incremental iteration, we will remove some nodes, hence $\mathcal{C}$ may be updated. Based on the way we construct the graph G, the initial $\mathcal{C}$ is $\mathcal{H}$. The weights of edges are *proximity scores* of two nodes, which is defined as $p(L_1, L_2) = \min_{s_r \in \mathcal{S}}(s_r \cdot \mathbb{1}_{s_r \geq 0.5})$ for any two 3D LSs L_1 and L_2. The $\mathbb{1}$ is an indicator function. The $s_r = e^{-(r/\tau_r)^2}$ is a normalized score measuring a distance r with a scaling factor τ_r. The $\mathcal{S}$ is the set of all normalized scores that measure the angle between the lines, the perpendicular distance from the endpoints to the other line, and the overlap ratio of the two LSs. These measurements are computed in both 3D space and 2D image space. Refer to supplementary material for details about τ_r and proximity scores. If the proximity score equals 0, we delete the corresponding edge. In graph theory,

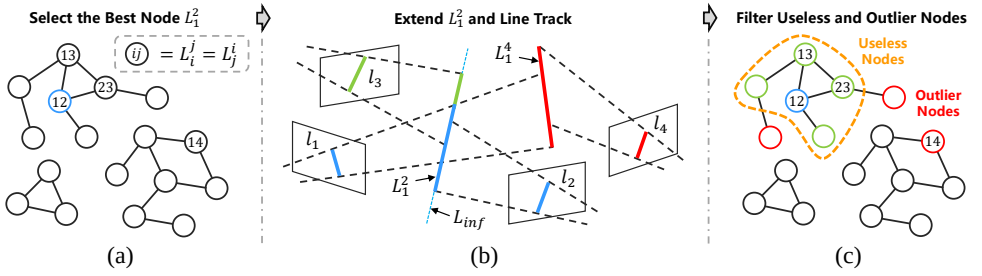


Figure 4: A sample iteration of incremental mapping. (a) The node (hypothesis) L_{12} has the highest strength score. (b) The L_{12} and line track are extended by matching graph under the reprojection verification. (c) The outlier and useless nodes are filtered from the graph.

the *strength* of a node is the sum of the weights of the edges attached to it [22]. In our case, the *strength* of L_i^j indicates the geometric supports that neighboring nodes provide to L_i^j. Thus, to evaluate L_i^j, we define its *strength* score as follows,

$$s(L_i^j) = \sum_{L_c \in \mathcal{C}} p(L_i^j, L_c). \tag{2}$$

Best Hypothesis Selection. In each iteration, we select the node with the highest strength score as the best node, as it receives the strongest geometric support from neighboring nodes and is thus most likely to be an inlier. The best node is then added to the existing 3D LS map, with its track elements being its two source 2D LSs. In the incremental process, we will filter out specific nodes in the graph. For robustness, if the best node lacks geometric support from at least two neighboring nodes, the incremental iteration is terminated.

Line Track Extension. Given the selected best node, assumed to be L_1^2 (refer to Fig. 4a), our goal is to extend it by adding more track elements, namely 2D LSs, to its line track. To accomplish this, we gather all 2D LSs that match l_1 or l_2 in a set $\{l_m\}$ and verify them using the reprojection of L_1^2. As shown in Fig. 4b, if $l_3 \in \{l_m\}$ passes the reprojection test, we add l_3 to the line track of L_1^2. We then project the two endpoints of l_3 back onto the 3D infinite line L_{inf} formed by L_1^2 to extend L_1^2. This process is repeated until no additional matched 2D LSs of the track elements pass the reprojection test.

Hypotheses Filtering. A 2D LS becomes a track element upon corresponding to a reconstructed 3D LS. After extending the line track, the nodes, where one source 2D LS is a track element and the other is not, are considered outliers. Since the two source 2D LSs come from a match but one source 2D LS does not pass the reprojection test in the Line Track Extension module, the match is an outlier and the node triangulated from this match is also an outlier. To prevent outlier nodes from providing geometric support to other nodes, *i.e.* impacting the best node selection in the next incremental iteration, we filter these outlier nodes from the graph. As shown in Fig. 4c, we delete node L_1^4 because l_4 does not pass the reprojection test in Fig. 4b. The outlier nodes may also belong to the same connected component as L_1^2. Furthermore, we eliminate nodes whose source 2D LSs are both track elements. Since these nodes fail to connect nodes where neither source 2D LS is a track element, they are useless for the next best node selection. In summary, we filter out nodes that have at least one source 2D LS identified as a track element, which is an efficient processing.

3.3 Post-Processing

Line Track Merging. To improve efficiency and robustness, we only match 2D LSs in neighboring images. However, such local matching might result in the same 3D LS entity being observed by multiple tracks but without matches between the respective track elements. To merge disconnected tracks, we first build a fully connected undirected weighted graph where the nodes are 3D LSs and the weights of the edges are 3D proximity scores of two nodes. Then, the edges are arranged in descending order based on their weights. Starting from the beginning of the edge list, two nodes of the edge are merged using the Union-Find algorithm [29]. Initially, the 3D LS of each group is the node itself. In each iteration, if the 3D proximity score of the 3D LS of two groups is greater than 0, we apply principal component analysis (PCA) to fit a merged 3D LS from the 3D LS of two groups. If all track elements of two groups pass the reprojection test against the merged 3D LS, we merge the group of the two nodes. The 3D LS and track of the merged group are updated by the merged 3D LS and track, respectively. The merging is repeated until no more 3D LSs can be merged.
Joint Optimization. Similar to the optimization module in LIMAP [19], we jointly optimize 3D LSs with SfM points and VPs. Specifically, the cost function comprises the summation of the reprojection error of SfM points and 3D LSs, the perpendicular distance between the associated SfM point and the 3D line, and the angle between the 3D direction of the associated VP and the 3D line. The 3D point-line and line-VP associations are based on 2D point-line and line-VP associations, point tracks, and line tracks.
Outlier Filtering. Finally, the track elements are filtered if they fail the reprojection test. The new 3D LS is the union of the back-projection of all inlier track elements onto the 3D infinite line formed by the original 3D LS. To enhance robustness, we only keep the 3D LS that has at least four visible images.

4 Experiments

We compare our method, IncreLM, with two state-of-the-art methods: L3D++[11] and LIMAP[19]. Our evaluation utilizes both the synthetic Hypersim dataset [26] and the real *Tanks and Temples* dataset [14]. The 2D LSs are produced from the LSD line detector [31] or the DeepLSD line detector [23]. The line feature matches are produced from GlueStick [24] line matcher. For all comparison methods, we use the same top 10 2D line matches.

4.1 Evaluation of 3D Line Mapping

Since there are no ground truth (GT) 3D LSs, we use the GT point cloud as the GT model. According to the evaluation pipeline of LIMAP [19], we use the following metrics to evaluate 3D LSs and line tracks. 1) *Length recall* (in meters) at τ ($R\tau$): sum of the lengths of the line portions within τ mm from the GT model. 2) *Inlier percentage* at τ ($P\tau$): the percentage of tracks that are within τ mm from the GT model. 3) *Average supports*: average number of image supports and 2D line supports across all line tracks. In addition to these metrics presented in LIMAP [19], we also evaluate the number of 3D LSs and the running times without considering feature detection and matching.

We conduct 3D line mapping on the first eight scenes of the Hypersim dataset [26]. The average metrics are reported in Tab. 1. Our method reconstructs the highest number of 3D LSs, achieves the greatest length recall, and secures the largest number of image

Line type	Method	R1	R5	R10	P1	P5	P10	# supports	# 3D lines	time [s]
LSD [31]	L3D++ [11]	31.0	167.4	222.1	**65.3**	**84.9**	**89.7**	17.2 / 19.9	545.6	2.0
	LIMAP [19]	37.1	208.7	282.5	59.3	80.5	86.2	16.8 / 19.7	692.6	55.5
	IncreLM	**39.3**	**228.5**	**310.7**	59.6	79.8	85.5	**18.9 / 23.1**	**850.6**	11.3
DeepLSD [23]	L3D++ [11]	39.1	189.9	249.1	**69.3**	**85.0**	**89.6**	16.1 / 18.6	490.2	1.8
	LIMAP [19]	32.6	181.0	237.6	61.0	79.2	84.2	**18.8 / 26.3**	512.3	53.6
	IncreLM	**50.5**	**270.8**	**358.4**	61.8	79.9	85.5	**18.8** / 21.9	**792.6**	10.8

Table 1: Line mapping results on the Hypersim dataset [26]. The notation $R\tau$ shows the length recall at τ mm and $P\tau$ shows the inlier percentage at τ mm.

Line type	Method	R5	R10	R50	P5	P10	P50	# supports	# 3D lines	time [s]
LSD [31]	L3D++ [11]	458.5	1056.7	3597.4	43.9	57.2	**86.9**	9.3 / 9.9	8979.3	30.0
	LIMAP [19]	601.7	1406.3	5007.8	42.6	55.6	85.8	8.9 / 9.9	10874.3	1304.5
	IncreLM	**716.5**	**1651.0**	**5696.9**	**45.7**	**57.7**	84.5	**17.5 / 20.8**	**14414.3**	283.8
DeepLSD [23]	L3D++ [11]	429.6	993.5	3415.4	44.9	**57.7**	**86.4**	9.1 / 9.8	7920.8	26.7
	LIMAP [19]	532.6	1248.3	4635.4	42.5	53.4	80.3	11.2 / 14.0	12050.5	1200.0
	IncreLM	**784.5**	**1820.6**	**6445.2**	**45.4**	57.0	83.1	**16.4 / 18.5**	**15443.2**	275.1

Table 2: Line mapping results on *train* split of *Tanks and Temples* [14]. The notation $R\tau$ shows the length recall at τ mm and $P\tau$ shows the inlier percentage at τ mm.

supports. It also maintains a competitive count of 2D line supports. This demonstrates that our method can produce more complete 3D LS maps and longer line tracks. Notably, L3D++ [11] achieves the fastest runtime speeds due to its use of GPU parallel computing and because it omits the processing of 3D points and VPs during the reconstruction phase. However, it falls short in terms of map completeness. It's worth mentioning that both our method and LIMAP [19] incorporate 3D points and VPs to enhance reconstruction quality. Our method outperforms LIMAP [19] in speed, demonstrating the efficiency of our approach.

The average metrics on the *train* split of *Tanks and Temples* are reported in Tab. 2. We remove the scene *Ignatius* as it has almost no line structures. Our method achieves the highest length recall, average support count, and number of 3D LSs, and operates noticeably faster than LIMAP [19]. Additionally, on LSD [31] lines, our method achieves the highest inlier percentage at 5mm and 10mm. This further demonstrates the superiority of our approach. The sample qualitative results on the testing data are displayed in Fig. 5. From this comparison, our method produces more complete 3D LS maps and is capable of reconstructing very dense 3D LSs, which will benefit many downstream computer vision tasks, such as visual localization [18] and surface reconstruction [15].

4.2 Evaluation of Incremental Mapping

We present the intermediate results of incremental mapping in Fig. 6. As iterations progress, the inlier percentage decreases while the length recall increases, indicating our method prioritizes the reconstruction of the precise 3D LSs. Consequently, the previous 3D line map provides valuable feedback for subsequent modeling, as filtered outlier nodes are more likely to represent true outliers. This mechanism is the cornerstone of our work, enhancing robustness against outlier matches. Leveraging an accurate 3D map to detect outliers and guide subsequent modeling helps yield more accurate and complete 3D LS maps.

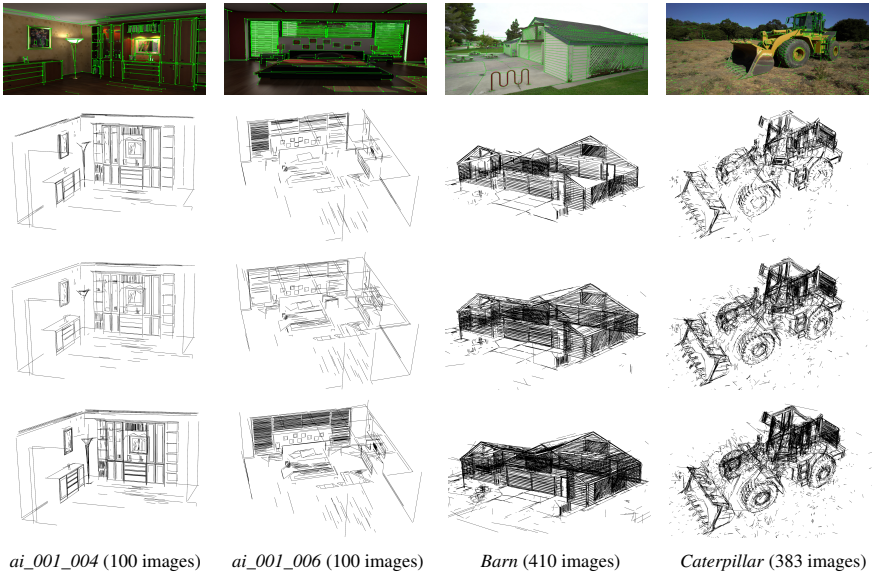


Figure 5: From top to bottom, the line mapping results are produced by L3D++ [11], LIMAP [19], and our method IncreLM with DeepLSD lines [23], respectively.

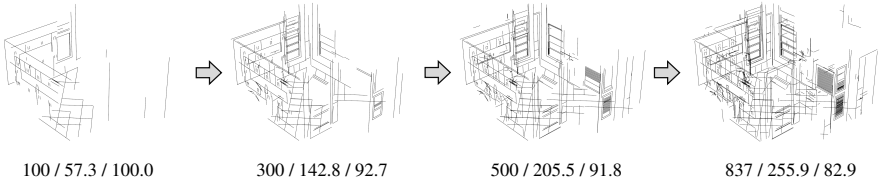


Figure 6: Line mapping results on a sample data of Hypersim [26]. The numbers below each subgraph display the number of reconstructed 3D LSs, length recall ($R5$), and inlier percentage ($P5$) sequentially. The last subgraph is the final 3D LS map.

Line-Line	Two-Points	Point-VP	R5	R10	R50	P5	P10	P50	# supports	# 3D lines	P_{lm}
✓			716.9	1653.1	5847.6	44.6	56.1	82.4	16.7 / 19.0	14387.3	100.0%
	✓		194.0	463.0	1550.3	**63.5**	**73.9**	**92.4**	**26.6 / 32.3**	1563.8	0.2%
		✓	409.5	971.6	3499.7	54.5	65.2	87.8	21.1 / 24.7	5420.5	0.6%
✓	✓	✓	**784.5**	**1820.6**	**6445.2**	45.4	57.0	83.1	16.4 / 18.5	**15443.2**	100.0%

Table 3: Ablation study of line triangulation solvers on *train* split of *Tanks and Temples* [14]. The notation $R\tau$ shows the length recall at τ mm and $P\tau$ shows the inlier percentage at τ mm. P_{lm} shows the percentage of used 2D line-line matches.

4.3 Ablation Studies

Line Triangulation. We conducted ablation studies on the solvers employed in hybrid RANSAC. The results are presented in Tab. 3, where a check mark indicates the use of a solver. Through this comparison, the individual use of any single solver fails to yield satisfactory results. The inlier percentage of the results obtained by the SfM point-related solvers, namely the Two-Points solver and the Point-VP solver, is high, but the length recall is low.

Method	R5	R10	R50	P5	P10	P50	# supports	# 3D lines	# supports*	# 3D lines*
IncreLM without merging	736.6	1701.1	6035.6	**45.4**	56.9	**83.1**	**16.4 / 18.6**	15150.1	7.3 / 8.2	46360.8
IncreLM with merging	**784.5**	**1820.6**	**6445.2**	**45.4**	**57.0**	**83.1**	**16.4** / 18.5	**15443.2**	7.8 / 8.8	44013.7

Table 4: Ablation study of line track merging on *train* split of *Tanks and Temples* [14]. The notation $R\tau$ shows the length recall at τ mm and $P\tau$ shows the inlier percentage at τ mm. The metrics with "*" are results before joint optimization. Otherwise, they are the final results.

Only 0.2% of line matches can utilize the Two-Points solver, and just 0.6% of line matches can utilize the Point-VP solver. This suggests that while the SfM points are reliable, they are too sparse for effective 3D line mapping. In contrast, the Line-Line solver exhibits a high length recall but a low inlier percentage, suggesting that the robust 3D line mapping is challenged by the one-to-many line matches inputs, which are heavily contaminated with outliers. By integrating these solvers into a unified hybrid RANSAC framework, the length recall of the produced results is improved, and the inlier percentage is also enhanced compared with using the Line-Line solver alone.

Line Track Merging. We conducted experiments with and without the line track merging module. As illustrated in Tab. 4, incorporating this module can increase the average track length before performing joint optimization. In the final results, merging operator results in more 3D LSs. This is because we merged 3D LSs supported by fewer than four images, which are vulnerable to being filtered out by the Outlier Filtering module, into 3D LSs supported by more images. The resulting merged 3D LSs are robust against such filtering. Although the individual Line Track Merging module reduces the number of 3D LSs, when combined with the Outlier Filtering module, these newly introduced 3D LSs increase the total number of final 3D LSs. The track length of the final results does not increase more because the track length of these newly introduced 3D LSs is usually short, thereby suppressing the growth of the average track length. Furthermore, incorporating the line track merging module significantly improves the length recall in the final results, demonstrating the effectiveness of our method.

5 Conclusion

In this paper, we introduce an incremental line mapping framework designed to address the challenge of numerous outliers. Our proposed 3D line segment hypotheses graph serves as a valuable tool for incremental mapping. Leveraging this graph, the previously established map provides feedback on subsequent scene modeling by filtering outlier nodes. Extensive real-world experiments validate the effectiveness and accuracy of our incremental mapping approach. Moving forward, our future work will focus on enhancing the efficiency of our framework through parallel programming.

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant Number: 2023YFB3906600), the National Natural Science Foundation of China (Grant Numbers: U22B2055, 62073320, 62273345, and U23A20386), and the Beijing Natural Science Foundation (Grant Number: L223003).

References

[1] 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] Qi Cai, Lilian Zhang, Yuanxin Wu, Wenxian Yu, and Dewen Hu. A pose-only solution to visual reconstruction and navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(1):73–86, 2021.

[3] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid camera pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 136–144, 2018.

[4] 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.

[5] Jianwei Guo, Yanchao Liu, Xin Song, Haoyu Liu, Xiaopeng Zhang, and Zhanglin Cheng. Line-based 3d building abstraction and polygonal surface reconstruction from images. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[7] 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.

[8] Manuel Hofer, Andreas Wendel, and Horst Bischof. Line-based 3d reconstruction of wiry objects. In *18th Computer Vision Winter Workshop*, pages 78–85, 2013.

[9] Manuel Hofer, Michael Maurer, and Horst Bischof. Improving sparse 3d models for man-made environments using line-based 3d reconstruction. In *IEEE International Conference on 3D Vision (3DV)*, volume 1, pages 535–542, 2014.

[10] Manuel Hofer, Michael Maurer, and Horst Bischof. Line3d: Efficient 3d scene abstraction for the built environment. In *German Conference on Pattern Recognition (GCPR)*, pages 237–248. Springer, 2015.

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

[12] Tong Hua, Tao Li, Liang Pang, Guoqing Liu, Wencheng Xuanyuan, Chang Shu, and Ling Pei. Plv-iekf: Consistent visual-inertial odometry using points, lines, and vanishing points. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–7, 2023.

[13] Arjun Jain, Christian Kurz, Thorsten Thormählen, and Hans-Peter Seidel. Exploiting global connectivity constraints for reconstruction of 3d line segments from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1586–1593, 2010.

[14] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[15] Pierre-Alain Langlois, Alexandre Boulch, and Renaud Marlet. Surface Reconstruction from 3D Line Segments. In *2019 International Conference on 3D Vision (3DV)*, pages 553–563. IEEE, 2019.

[16] Xin Li, Yijia He, Jinlong Lin, and Xiao Liu. Leveraging planar regularities for point line visual-inertial odometry. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 5120–5127, 2020.

[17] Hyunjun Lim, Jinwoo Jeon, and Hyun Myung. Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping. *IEEE Robotics and Automation Letters (RAL)*, 7(2):1518–1525, 2022.

[18] Hongmin Liu, Chengyang Cao, Hanqiao Ye, Hainan Cui, Wei Gao, Xing Wang, and Shuhan Shen. Lightweight structured line map based visual localization. *IEEE Robotics and Automation Letters (RAL)*, 9(6):5182–5189, 2024.

[19] Shaohui Liu, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. 3d line mapping revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21445–21455, 2023.

[20] Lalit Manam and Venu Madhav Govindu. Correspondence reweighted translation averaging. In *European Conference on Computer Vision (ECCV)*, pages 56–72. Springer, 2022.

[21] Branislav Micusik and Horst Wildenauer. Structure from motion with line segments under relaxed endpoint constraints. *International Journal of Computer Vision (IJCV)*, 124:65–79, 2017.

[22] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks*, 32(3):245–251, 2010.

[23] Rémi Pautrat, Daniel Barath, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Deeplsd: Line segment detection and refinement with deep image gradients. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17327–17336, 2023.

[24] Rémi Pautrat, Iago Suárez, Yifan Yu, Marc Pollefeys, and Viktor Larsson. Gluestick: Robust image matching by sticking points and lines together. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9706–9716, 2023.

[25] Long Quan and Takeo Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(8):834–845, 1997.

[26] 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 *IEEE International Conference on Computer Vision (ICCV)*, pages 10912–10922, 2021.

[27] Yohann Salaün, Renaud Marlet, and Pascal Monasse. Line-based robust SfM with little image overlap. In *IEEE International Conference on 3D Vision (3DV)*, pages 195–204, 2017.

[28] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[29] Robert Sedgewick and Kevin Wayne. *Algorithms*. Addison-wesley professional, 2011.

[30] Fangwen Shu, Jiaxuan Wang, Alain Pagani, and Didier Stricker. Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2105–2112, 2023.

[31] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(4):722–732, 2008.

[32] Runzhi Wang, Kaichang Di, Wenhui Wan, and Yongkang Wang. Improved point-line feature based visual slam method for indoor scenes. *Sensors*, 18(10):3559, 2018.

[33] Dong Wei, Yi Wan, Yongjun Zhang, Xinyi Liu, Bin Zhang, and Xiqi Wang. Elsr: Efficient line segment reconstruction with planes and points guidance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15807–15815, 2022.

[34] Juyang Weng, Thomas S Huang, and Narendra Ahuja. Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(03):318–336, 1992.

[35] Kuan Xu, Yuefan Hao, Shenghai Yuan, Chen Wang, and Lihua Xie. Airvo: An illumination-robust point-line visual odometry. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3429–3436, 2023.

[36] Zewen Xu, Hao Wei, Fulin Tang, Yidi Zhang, Yihong Wu, Gang Ma, Shuzhe Wu, and Xin Jin. Plpl-vio: a novel probabilistic line measurement model for point-line-based visual-inertial odometry. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 5211–5218, 2023.

[37] Haozhi Yang, Jing Yuan, Yuanxi Gao, Xingyu Sun, and Xuebo Zhang. Uplp-slam: Unified point-line-plane feature fusion for rgb-d visual slam. *Information Fusion*, 96: 51–65, 2023.

[38] 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.

[39] Lipu Zhou, Jiacheng Liu, Fengguang Zhai, Pan Ai, Kefei Ren, Yinian Mao, Guoquan Huang, Ziyang Meng, and Michael Kaess. Efficient bundle adjustment for coplanar points and lines. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8356–8363, 2023.