

# IncreLM: Incremental 3D Line Mapping (Supplementary Material)

Xulong Bai<sup>1,2,3</sup>  
 baixulong2022@ia.ac.cn  
 Hainan Cui<sup>\*1,2,3</sup>  
 hncui@nlpr.ia.ac.cn  
 Shuhan Shen<sup>\*1,2,3</sup>  
 shshen@nlpr.ia.ac.cn

<sup>1</sup> Institute of Automation,  
 Chinese Academy of Sciences,  
 Beijing, China

<sup>2</sup> School of Artificial Intelligence,  
 University of Chinese Academy of  
 Sciences, Beijing, China

<sup>3</sup> CASIA-SenseTime Research Group,  
 Beijing, China

In this document, we present supplementary material for our main paper. The implementation details are provided in Section A and the experimental details are provided in Section B. In Section C, we compare the performance of line-assisted visual localization using 3D LS maps generated by our method versus those generated by LIMAP [9].

## A Implementation Details

### A.1 Reprojection Score and Reprojection Test

In our approach, we conduct reprojection tests across various modules to improve geometric consistency across multiple views. The reprojection test utilizes a reprojection score. Let  $L$ ,  $l$ , and  $\pi(L)$  represent a 3D LS, a 2D LS, and the 2D LS projection of  $L$  onto the image of  $l$ , respectively. The reprojection score is calculated as follows,

$$s_{re}(l, \pi(L)) = \min_{s_r \in \mathcal{S}_{re}} (s_r \cdot \mathbb{1}_{s_r \geq 0.5}), \quad (1)$$

where  $s_r$  represents a normalized score quantifying a specific type of distance  $r$ . The set  $\mathcal{S}_{re} = \{s_{a_{2D}}, s_{p_{2D}}, s_{o_{2D}}\}$  comprises three normalized scores. These normalized scores respectively measure the angle  $a_{2D}$ , the endpoint perpendicular distance  $p_{2D}$ , and the overlap ratio  $o_{2D}$ . Specifically, the angle  $a_{2D}$  is the angle between  $l$  and  $\pi(L)$ . The endpoint perpendicular distance  $p_{2D}$  is defined as the maximum perpendicular distance from the endpoints of  $l$  to the 2D infinite line formed by  $\pi(L)$ . To determine the overlap ratio  $o_{2D}$ ,  $\pi(L)$  is orthogonally projected onto  $l$ . The endpoints of this projection are clipped if they extend beyond  $l$ , resulting in a 2D LS  $\Pi(\pi(L))$ . The ratio  $o_{2D}$  is then calculated as the length of  $\Pi(\pi(L))$  divided by the length of  $l$ , i.e.,  $o_{2D} = |\Pi(\pi(L))|/|l|$ . Both  $s_{a_{2D}}$  and  $s_{p_{2D}}$  are calculated using the formula  $s_r = e^{-(r/\tau_r)^2}$ , where  $\tau_r$  is a scaling factor for the distance  $r$ . The scaling factors are set as  $\tau_{a_{2D}} = 5$  degrees,  $\tau_{p_{2D}} = 2$  pixels. The  $s_{o_{2D}}$  is set to 1 if  $o_{2D} > 0$ ; otherwise, it is set to 0. If the reprojection score exceeds 0, it indicates that the reprojection test is passed.

\*Corresponding author

## A.2 Proximity Score

Let  $L_1$  and  $L_2$  denote two 3D LSs, the proximity score  $p(L_1, L_2)$  is calculated by combining the 2D proximity score  $p_{2D}(L_1, L_2)$  with the 3D proximity score  $p_{3D}(L_1, L_2)$ , and is expressed as follows,

$$p_{2D}(L_1, L_2) = \min_{s_r \in \mathcal{S}_{2D}} (s_r \cdot \mathbb{1}_{s_r \geq 0.5}) \quad (2)$$

$$p_{3D}(L_1, L_2) = \min_{s_r \in \mathcal{S}_{3D}} (s_r \cdot \mathbb{1}_{s_r \geq 0.5}) \quad (3)$$

$$p(L_1, L_2) = \min(p_{2D}(L_1, L_2), p_{3D}(L_1, L_2)) = \min_{s_r \in \mathcal{S}} (s_r \cdot \mathbb{1}_{s_r \geq 0.5}) \quad (4)$$

$$\mathcal{S} = \mathcal{S}_{2D} \cup \mathcal{S}_{3D} \quad (5)$$

where  $s_r$  represents a normalized score quantifying a specific type of distance  $r$ . As discussed in Section 1 of our main paper, each *line track* is a set of 2D LSs in images, corresponding to certain parts of the same 3D LS entity. We assume  $\mathcal{T}_1$  and  $\mathcal{T}_2$  are the line tracks corresponding to  $L_1$  and  $L_2$ , respectively. In particular, if  $L_1$  and  $L_2$  are two 3D LS hypotheses,  $\mathcal{T}_1$  and  $\mathcal{T}_2$  are defined as their respective source 2D LSs. Let  $\mathcal{T}_2 \setminus \mathcal{T}_1$  represent the set of elements that remain after removing the elements of  $\mathcal{T}_1$  from set  $\mathcal{T}_2$ . Similarly, let  $\mathcal{T}_1 \setminus \mathcal{T}_2$  represent the set of elements that remain after removing the elements of  $\mathcal{T}_2$  from set  $\mathcal{T}_1$ . To calculate the 2D proximity score, for each element (2D LS) in the set  $\mathcal{T}_2 \setminus \mathcal{T}_1$ , we calculate the reprojection scores of  $L_1$ , and similarly for each element (2D LS) in the set  $\mathcal{T}_1 \setminus \mathcal{T}_2$ , we calculate the reprojection scores of  $L_2$ . These reprojection scores are defined by Eq. (1). The 2D proximity score,  $p_{2D}(L_1, L_2)$ , is determined by the minimum of all these reprojection scores. To maintain a consistent format with  $p_{3D}(L_1, L_2)$ , we represent  $p_{2D}(L_1, L_2)$  using the set of normalized scores. The set  $\mathcal{S}_{2D}$  encompasses all normalized scores derived from these reprojection calculations. Let  $m$  and  $n$  be the numbers of elements in the sets  $\mathcal{T}_2 \setminus \mathcal{T}_1$  and  $\mathcal{T}_1 \setminus \mathcal{T}_2$ , respectively. The total number of reprojection scores involved is  $m + n$ , and the total number of normalized scores in  $\mathcal{S}_{2D}$  is  $3m + 3n$ . This is because each reprojection score involves three normalized scores, namely  $s_{a_{2D}}$ ,  $s_{p_{2D}}$  and  $s_{o_{2D}}$ , as defined in Eq. (1).

Similar to  $\mathcal{S}_{re}$ , as defined in Eq. (1),  $\mathcal{S}_{3D} = \{s_{a_{3D}}, s_{p_{3D}}, s_{o_{3D}}\}$  comprises three normalized scores that measure the angle  $a_{3D}$ , the scale-invariant endpoint perpendicular distance  $p_{3D}$ , and the overlap ratio  $o_{3D}$ , respectively. Specifically, the angle  $a_{3D}$  is the angle between  $L_1$  and  $L_2$ . Defining  $p_{3D}$  as the maximum perpendicular distances between the endpoints of  $L_1$  and  $L_2$  to each other's line will impose a penalty on longer 3D LS. However, these long 3D LSs are important for a clean reconstruction. To address this issue, we unproject  $L_1$  and  $L_2$  onto each other's line to get two 3D LS projections. We calculate  $p_{3D}$  as the maximum perpendicular distances between the endpoints of the 3D LS projections to each other's 3D LS projection, similar to the approach used in LIMAP [9]. To obtain scale invariance, we divide  $p_{3D}$  by  $\min(\sigma_1, \sigma_2)$ , where  $\sigma_i = \text{median}(\sigma_i^k)$ ,  $\sigma_i^k = d_i^k / f_i^k$ ,  $d_i^k$  is the depth of the midpoint of two 3D endpoints, back-projected from the 2D endpoints of the  $k$ th 2D LS observation to the corresponding 3D LS  $L_i$ , and  $f_i^k$  is the focal length of the  $k$ th 2D LS observation's image. The overlap ratio  $o_{3D}$  is determined by projecting  $L_1$  and  $L_2$  onto each other and calculating the maximum length ratio between the projections and the original 3D LSs. The  $s_{a_{3D}}$  and  $s_{p_{3D}}$  are calculated using the formula  $s_r = e^{-(r/\tau_r)^2}$ , where  $\tau_r$  is a scaling factor for the distance  $r$ . The scaling factors are set as  $\tau_{a_{3D}} = 5$  degrees,  $\tau_{p_{3D}} = 1$ . The  $s_{o_{3D}}$  is set to 1 if  $o_{3D} > 0$ ; otherwise it is set to 0. The final set  $\mathcal{S}$ , which combines all the normalized scores, is the union of  $\mathcal{S}_{2D}$  and  $\mathcal{S}_{3D}$ , encapsulating the comprehensive scoring framework for the proximity measurement between  $L_1$  and  $L_2$ .

### A.3 Joint Optimization

To improve 3D line mapping quality, we jointly optimize 3D lines with SfM points and VPs by minimizing the following energy function [9],

$$E = \sum_P E_P(P) + \sum_L E_L(L) + \sum_{(P,L)} E_{PL}(P,L) \quad (6)$$

where  $E_P$  is the squared point reprojection error,  $E_L$  is the squared line reprojection error, and can be written as follows,

$$E_L(L) = \sum_k w_{\angle}^2(L_k, l_k) \cdot e_{\text{perp}}^2(L_k, l_k), \quad (7)$$

$$w_{\angle}(L_k, l_k) = \exp(\alpha(1 - \cos(\angle(L_k, l_k))), \quad (8)$$

where  $l_k$  is the detected 2D LS,  $L_k$  is the 2D infinite line projection of the 3D LS  $L$  on the image of  $l_k$ , and  $e_{\text{perp}}(L_k, l_k)$  is the root of the sum of the squared perpendicular distance from the endpoints of  $l_k$  to  $L_k$ .  $\alpha$  equals 10.0 in our system. The function  $E_{PL}(P, L)$  encodes the perpendicular distance between the associated SfM point and the 3D line, the angle between the 3D direction of the associated VP and the 3D line, and VP orthogonality regularization. The associated SfM points of the 3D LS are derived from 2D point-line associations, SfM point tracks, and line tracks. Similarly, the associated VPs of the 3D LS are derived from 2D line-VP associations and line tracks.

## B Experimental Details

### B.1 Datasets

We utilize the Hypersim dataset [9] and the *Tanks and Temples* dataset [2] for experiments.

For Hypersim dataset [9], we utilize images from the first eight scenes for our experiments, resizing each to a maximum dimension of 800 pixels. The SfM points are derived using COLMAP [2] with the provided Ground Truth (GT) camera poses. The evaluation model (GT points) is constructed from GT depth maps and GT poses using the LIMAP library [9].

For *Tanks and Temples* dataset [2], we utilize images from the training data for our experiments, excluding the scene *Ignatius* as it has almost no line structures. The evaluation model (GT points) is the provided GT point cloud. The SfM points and camera poses are derived using COLMAP [2]. We align the camera poses with the provided GT point cloud for accurate evaluation. Since the provided GT point cloud primarily focuses on the main object, and our method is capable of reconstructing 3D LSs far from the main object, we take measures to ensure distant 3D LSs do not compromise reconstruction accuracy. Following LIMAP’s recommendations, we compute an axis-aligned bounding box around the GT points, extend it by 0.1 meters in all three dimensions, and restrict our evaluations to the lines within this expanded region.

### B.2 Details on L3D++ [10] and LIMAP [9]

We compare our method with two state-of-the-art methods: L3D++ [10] and LIMAP [9], utilizing their publicly available source code. For L3D++ [10], we use their default parameters.

Scene	HLoc [8]	LIMAP [9]	IncreLM
Chess	<b>2.4</b> / 0.84 / 93.0	2.5 / 0.85 / 92.3	2.5 / <b>0.83</b> / <b>93.2</b>
Fire	2.3 / 0.89 / 88.9	2.1 / 0.84 / 95.5	<b>1.9</b> / <b>0.77</b> / <b>97.1</b>
Heads	<b>1.1</b> / <b>0.75</b> / <b>95.9</b>	<b>1.1</b> / 0.76 / <b>95.9</b>	<b>1.1</b> / <b>0.75</b> / 94.6
Office	3.1 / 0.91 / 77.0	<b>3.0</b> / 0.89 / 78.4	<b>3.0</b> / <b>0.86</b> / <b>80.2</b>
Pumpkin	5.0 / 1.32 / 50.4	4.7 / 1.23 / 52.9	<b>4.5</b> / <b>1.18</b> / <b>56.0</b>
Redkitchen	4.2 / 1.39 / 58.9	4.1 / 1.39 / 60.2	<b>3.9</b> / <b>1.35</b> / <b>62.9</b>
Stairs	5.2 / 1.46 / 46.8	<b>3.7</b> / 1.02 / 71.1	<b>3.7</b> / <b>0.97</b> / <b>73.3</b>
Avg.	3.3 / 1.08 / 73.0	3.0 / 1.00 / 78.0	<b>2.9</b> / <b>0.96</b> / <b>79.6</b>

Table 1: Per-scene results of visual localization on 7Scenes [8]. We report the median translation and rotation error in cm and degrees, along with the percentage of the poses at a 5cm / 5deg threshold, presented sequentially.

For LIMAP [9], we employ the Line + Line, Multiple Points, Line + Point, and Line + VP methods for two-view line triangulation, employ the default *greedy* strategy for generating line tracks and conduct joint optimization using SfM points, 3D lines and VPs.

For the sake of fairness, all methods receive identical inputs: detected 2D LSs, VPs, and top  $K$  2D line matches, where  $K = 10$ . These 2D line matches are supplied by the state-of-the-art line matcher GlueStick [9]. Similar to the evaluation framework used by LIMAP [9], we evaluate 3D LS with at least four supporting images. All experiments are conducted on a computer equipped with a 3.4GHz processor.

## C Evaluation on Line-assisted Visual Localization

We compare the performance of line-assisted visual localization using 3D LS maps generated from LIMAP [9] against those produced by our proposed method, IncreLM. Specifically, we conduct the experimental evaluation using the visual localization framework of LIMAP [9] on the 7Scenes dataset [8]. The results are presented in Table 1. In addition, we reported results from the point-based visual localization method, HLoc [8].

As depicted in Table 1, our line-assisted visual localization consistently outperforms the point-based approach, which demonstrates integrating point and line features significantly enhances visual localization accuracy. Furthermore, utilizing our 3D LS maps for line-assisted visual localization yields superior performance compared to the 3D LS maps from LIMAP [9]. This demonstrates that our method produces more precise 3D LS maps and again emphasizes our approach’s advantages.

## References

- [1] Manuel Hofer, Michael Maurer, and Horst Bischof. Efficient 3d scene abstraction using line segments. *Computer Vision and Image Understanding (CVIU)*, 157:167–178, 2017.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12716–12725, 2019.
- [7] 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.
- [8] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013.