ISPRS Journal of Photogrammetry and Remote Sensing 210 (2024) 80–96

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

The One-Point-One-Line geometry for robust and efficient line segment correspondence

Haoyu Guo [a], Dong Wei [b,*], Yongjun Zhang [b,*], Yi Wan [b], Zhi Zheng [c], Yongxiang Yao [b], Xinyi Liu [b], Zhuofan Li [d]

[a] Institute of Water Engineering Sciences, Wuhan University, Wuhan 430079, China
[b] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[c] Institute of Space and Earth Information Science, The Chinese University of Hong Kong, Shatin, N.T., China
[d] College of Tourism, Xinyang Normal University, Xinyang 464000, China

ARTICLE INFO

Keywords:
Line matching
Epipolar geometry
Point orientation
One-point-one-line geometry
3D reconstruction

ABSTRACT

Three-dimensional (3D) lines are common elements in artificial scenes and serve as basic, yet essential features for structural 3D reconstruction. The crucial step of 3D line reconstruction, namely two-view line segment matching, still faces challenges in terms of both accuracy and efficiency improvements. Therefore, robust and efficient constraints are needed to establish valid line candidates. This paper introduces a novel geometry constraint called "one-point-one-line geometry" (OPOL) to enhance the precision of line matching and reduce computational complexity. OPOL offers two remarkable advantages: (1) It takes point orientations as the constraint, which is not only invariant to projective transformations, but also alleviates computational requirements. (2) It needs only one point match to construct the geometry constraint, thus both the grouping and validation are greatly reduced. Additionally, we incorporate the line sweep strategy into OPOL, leveraging depth and space constraints derived from existing 3D points to further enhance efficiency. Extensive experiments conducted on large-coverage and high-resolution images (as large as 10336 × 7788 pixels) demonstrated that OPOL matched lines within a second for an image pair. Both quantitative and qualitative results also demonstrated the superior accuracy and efficiency performance of OPOL. We integrated OPOL into multiple view line reconstruction frameworks, and the promising experimental results reveal the performance of OPOL for robust line reconstruction. The OPOL code is publicly available at https://github.com/JoeAlexxxxx/OPOL.

1. Introduction

Three-dimensional (3D) lines containing rich structural details are prominent features in artificial scenes (Hofer et al., 2015). Line segment matching, serving as the fundamental step in recovering 3D lines from two-view or multiple view images, is essential for various aspects of 3D reconstruction, including structure recovery, plane reconstruction, and scene abstraction. Line matching is much more complicated than traditional point matching due to the challenging trade-offs between robust matching descriptors and low matching complexity (Lange et al., 2019, Vakhitov et al., 2019, Lange et al., 2020). The description is weak when the texture along the line region is poor. Although introducing scene plane geometry could enhance the robustness of matching descriptors, the encountered feature grouping operations for establishing geometry constraint would greatly prolong the processing time and increase algorithms complexity. Efficient and accurate line segment matching is still on the way.

Researchers have dedicated efforts for line matching with joint features, such as joint points and lines. They first recovered the local transformation and then checked whether the line candidate were aligned with the transformation. Fan et al. (2010) used four points to obtain local projections, which had a complexity of $o(n^5)$ for the whole matching. When the relative pose of two-view images is known, three points are generally required to calculate the local projection, and thus the time complexity reaches to $o(n^4)$. Ramalingam et al. (2015) accelerated the validation of line matches with two points and two lines, and the time complexity was reduced to $o(n^3)$ with the line sweep strategy. Wei et al. (2022) used two junction lines to obtain and validate the local

* Corresponding authors.
E-mail addresses: haoyu.guo@whu.edu.cn (H. Guo), weidong@whu.edu.cn (D. Wei), zhangyj@whu.edu.cn (Y. Zhang), yi.wan@whu.edu.cn (Y. Wan), zhizheng@cuhk.edu.hk (Z. Zheng), yaoyongxiang@whu.edu.cn (Y. Yao), liuxy0319@whu.edu.cn (X. Liu), lizhuofan@xynu.edu.cn (Z. Li).

https://doi.org/10.1016/j.isprsjprs.2024.03.003
Received 16 August 2023; Received in revised form 31 January 2024; Accepted 4 March 2024
Available online 13 March 2024

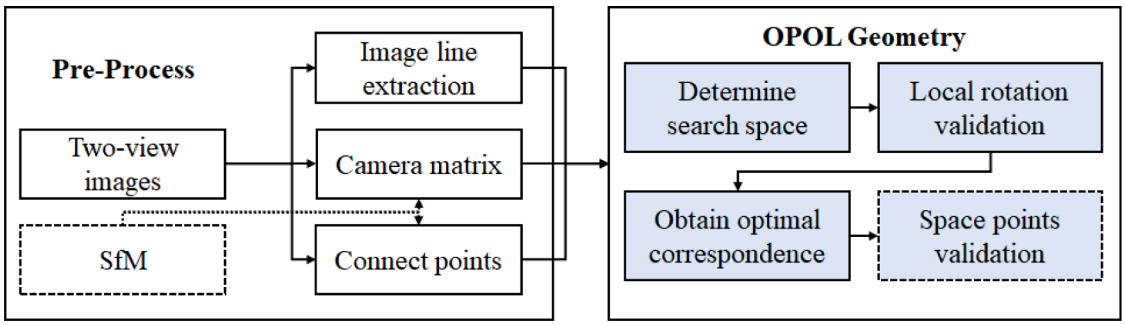


Fig. 1. The pipeline of OPOL.

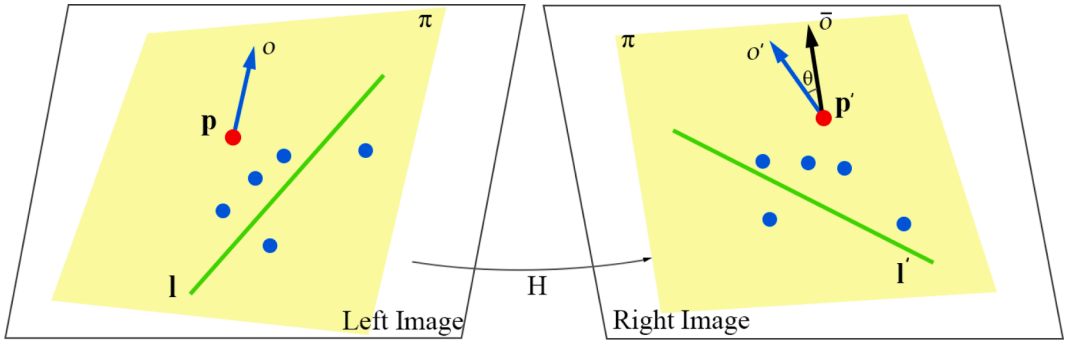


Fig. 2. Illustration of one-point-one-line geometry. $l \leftrightarrow l'$ is the line candidate need to be checked. $p \leftrightarrow p'$ is a pair of point matches outside the line and containing the main directions (o and o') obtained from the local descriptor. The local homography H can be calculated with $l \leftrightarrow l'$ and $p \leftrightarrow p'$, with which we could compute the rotation of o in the second image as $\bar{o}$. Finally, $l \leftrightarrow l'$ can be validated via the orientation difference θ.

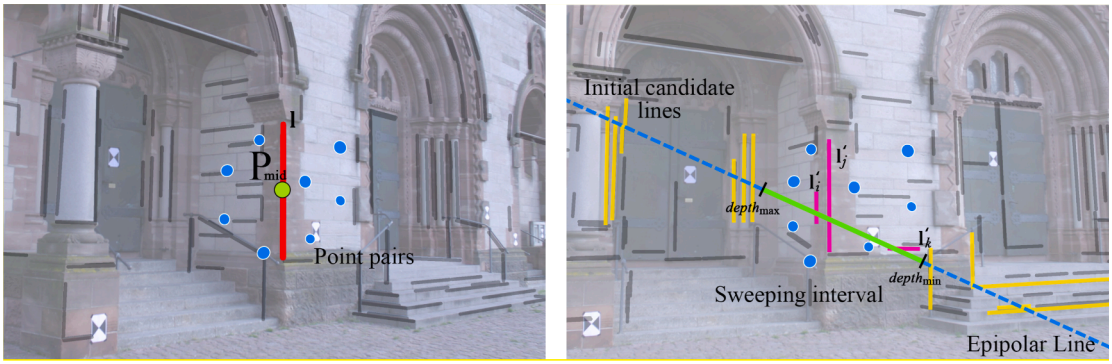


Fig. 3. Illustration of line sweeping. With the middle point of the red line in the left image, we could sweep its candidate along the epipolar line by pre-building the line map of the right image. Also, the depth of the neighbor points of p_{mid} is used to constrain the interval of the line sweeping. Finally, the candidates of l are constrained in the red interval to be l'_i to l'_k.

Table 1
The image datasets in each section.

Section	Dataset	Image Number	Image Size(pix)	Two-view pairs
5.1	Herze-Jesu	25	3072 × 2048	75
	Castle	30	3072 × 2048	87
	Rathaus	84	5184 × 3456	252
5.2	Dublin	368	9000 × 6732 7360 × 4912	1104
	Guangzhou	419	10336 × 7788	1235

projection, but it still has the complexity of $o(n^3)$. For this kind of matching paradigm, the time complexity closely depends on the number of features required to establish the geometry constraint. The fewer joint features that are required, the more efficient the matching will be. Thus, researchers always consider the accurate geometry constraint with fewer joint features.

Generally, point orientations could have been obtained in advance in many SfM (Structure from Motion) (Wu et al., 2013, Gong et al., 2023) or SLAM (Simultaneous Localization and Mapping) tasks. While it is usually neglected in constructing the geometry constraint of line matching. In this paper, we exploit point orientation and propose a novel one-point-one-line (OPOL) constraint to establish line correspondence. The key idea of OPOL is simple while effective: if a point and line lie in the same plane of the object space, they have the same projection in two image planes; also, the main orientation of the two key points should be aligned with the transformation. This basic hypothesis requires only one point match to validate the line pair, thus benefits the time complexity to $o(n^2)$. The pipeline of line matching with OPOL is shown in Fig. 1. OPOL uses the existing point match, of which the

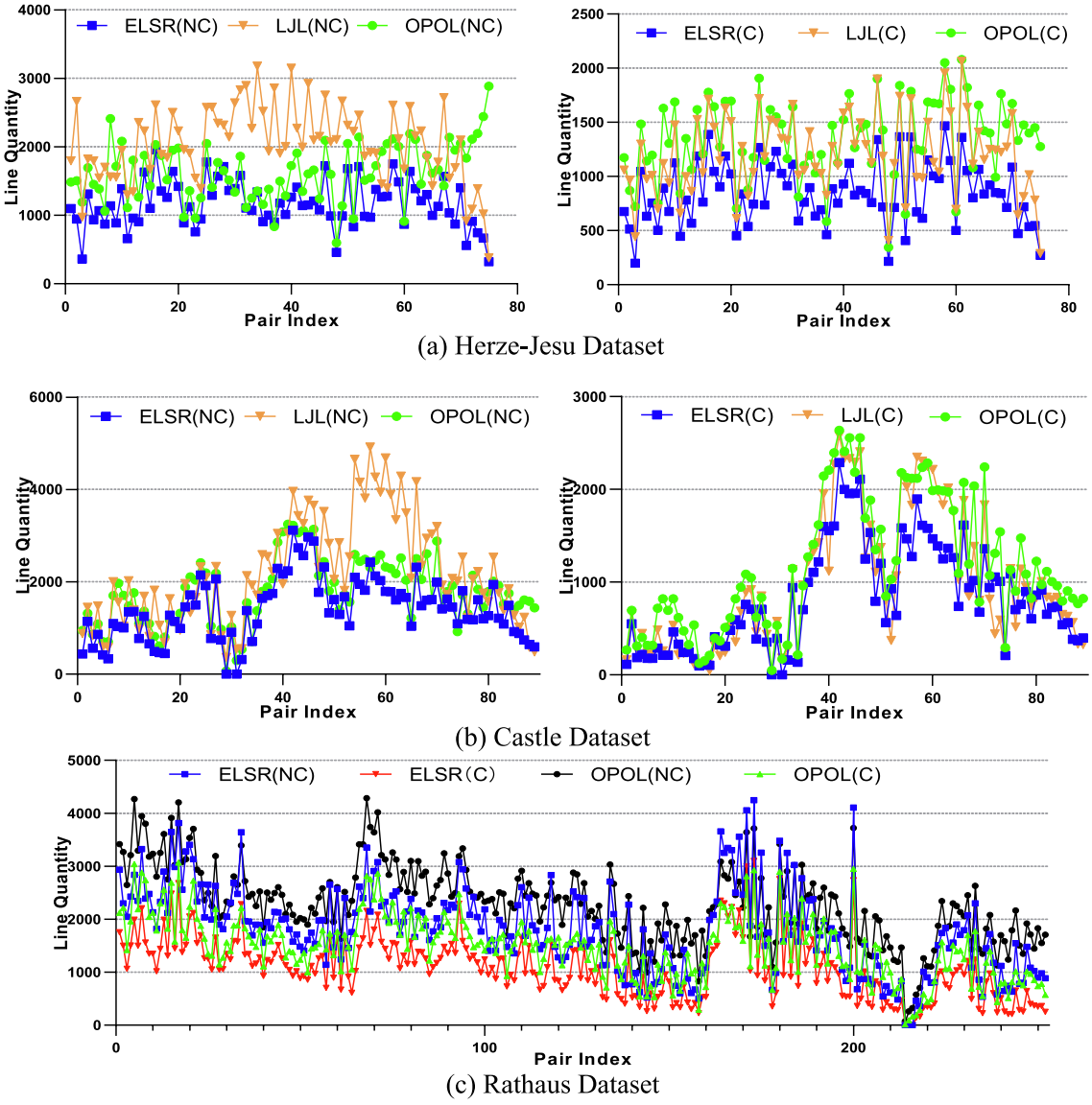


Fig. 4. The line matching results of two-view ground images. For all of the experiments, (C) means that the algorithm is controlled with dense points; (NC) means that no control is employed. Note that in the following experiments, we use dense points as ground truth to validate the correctness of the result.

matching method has been quite robust. Inspired by (Barath et al., 2018), OPOL introduces the point orientation to construct a reliable line matching geometry. Furthermore, the depth constraint of neighbor points is used to accelerate the search for line candidates by shrinking the epipolar line trace.

Three common properties in photogrammetry contribute to the effectiveness of OPOL. First, in artificial scenes, line segments are frequently found inside planes and are easy to be coplanar with neighboring points. Second, the main orientations of point features in the plane are aligned under the projective transformation (Lowe et al., 2004). Third, OPOL can utilize point clouds such as sparse points, dense points, or LiDAR points, which have been readily available in many applications. The experiment with thousands of image pairs in large size would present two notable advantages of OPOL: (1) it is quite fast due to its low complexity and few grouping processes; (2) OPOL can obtain more line matches because fewer joint features have the higher chance to be coplanar while it only requires one neighbor point for validation.

This paper is organized as follows. Section 2 provides a brief overview of line segment matching. In Section 3, we present OPOL geometry constraints for corresponding line segments, which leverages the information derived from points and their orientations. The process of filtering candidate matches using a line-sweep strategy is also elucidated within the same section. The algorithmic complexity of OPOL is analyzed in Section 4. Experiments are described in Section 5. Finally, Section 6 concludes the article.

2. Related works

2.1. Matching with textures

Like point matching, many line-matching algorithms used texture along line segments to construct the line descriptor and establish line matches. Bay et al. (2005) built the gradient histogram to obtain initial candidates, and a topological filter was used to remove mismatches. Wang et al. (2009) proposed a line descriptor constructed by mean-standard deviation (MSLD) of the texture matrix. But it was sensitive to the affine transformation. Some studies have been conducted to address the change in viewpoint. Zhang and Koch (2013) proposed the line band descriptor (LBD) that incorporated local textures and geometric constraints. It could overcome the affine transformations to a certain degree. López et al. (2015) constructed the descriptor with multiscale information and introduced it into the matching with

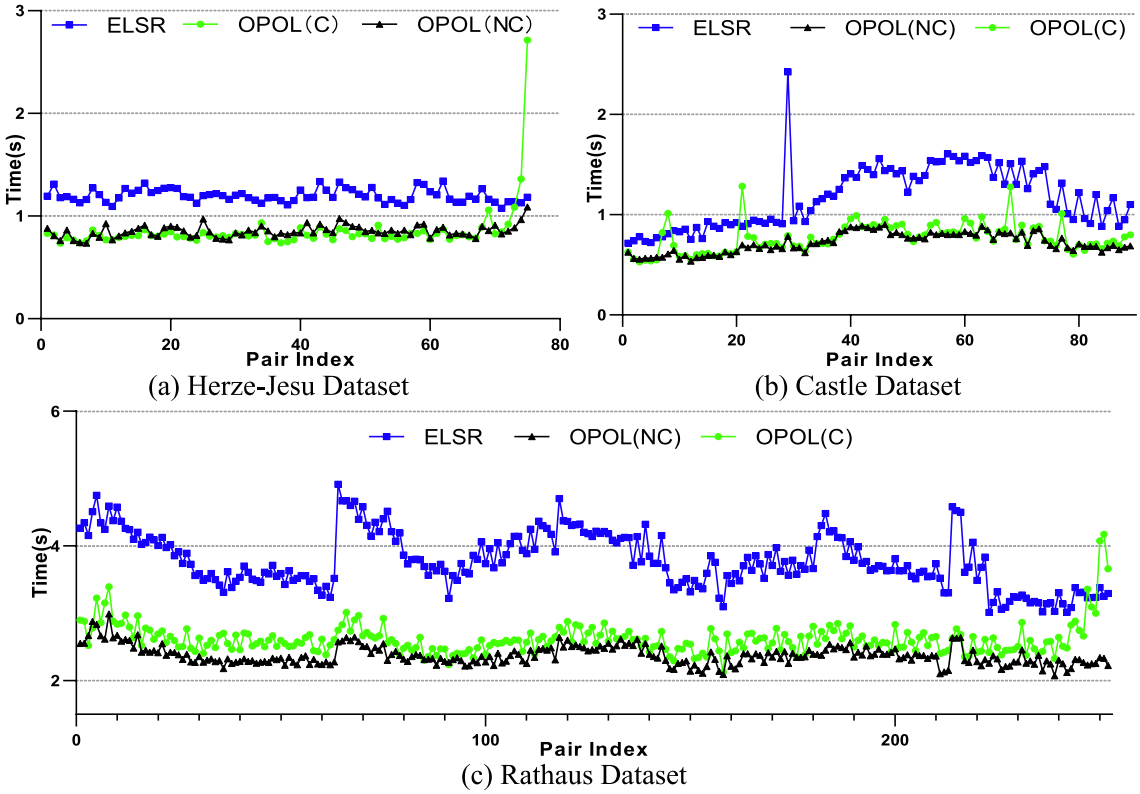


Fig. 5. The time consumption of ground images. All the methods were run in a single thread. LJL was not plotted because it was much slower than OPOL and ELSR.

geometry constraint. Li et al. (2016) introduced the line-junction-line (LJL) structure composed of junction lines and intersection points, within which the orientation histograms were grouped as the descriptor, and the multiscale pyramid was also built to counter the scale change. Li and Yao (2017) then improved LJL to be affine and scale invariant by recovering the affine transformation with the junction lines and their intersections, in which the image pyramid was no longer required.

Recently, learning-based line matching methods (Lange et al., 2019, Vakhitov et al., 2019, Lange et al., 2020, Abdellali et al., 2021), such as SOLD2 (Pautrat et al., 2021) and L2D2 (Abdellali et al., 2021), have been proposed with the rapid development of deep learning. These methods could extract and describe the line simultaneously and showed great potential in line matching. But these methods also had the common drawbacks of deep learning methods, e.g., the low efficiency for large size images and the limitation of the transferability for different scenes.

2.2. Matching with geometry constraint

Relying on texture descriptors alone will disregard the geometry information of line segments. Thus, many researchers introduced geometry constraints into line matching. Zhang et al. (2002) employed epipolar geometry to limit the search space to a quadrilateral, which greatly accelerated the candidate searching. Ok et al. (2012a) proposed the similarity measurement based on the daisy descriptor (Tola et al., 2009) to establish the correspondence. In addition, a combination of epipolar, radiometric, correlation, and regional constraints is used to obtain a reliable match. Al-Shahri et al. (2014) combined global geometric constraint with local texture validation, which employed epipolar constraint to identify candidate matches and then utilized local line configuration and transformation to reduce geometric ambiguities. Chen et al. (2021) classified line segments into structured line pairs, unstructured line pairs, and individual line pairs, with which the hierarchical geometry constraint was used to overcome the viewpoint change. Wang et al. (2021) matched line segments with line pairs and constructed the double-layer matrix through collinear geometry to improve matching accuracy. Some studies (Schmid et al., 1997) also improved texture correlation with epipolar geometries, in which the local homography was calculated for texture distortion.

Due to advanced development in the extraction and matching of point features, some studies have exploited point matches to assist line matching. Fan et al. (2010) and Sun et al. (2015) grouped joint points to obtain the local projection to validate line matching. Jia et al. (2019) constructed the topological adjacency between a point and a line, with which the candidate could be filtered and overall performance was improved. Furthermore, Shi et al. (2017) used the mesh topology to speed up the matching. Wang et al. (2020) used camera poses and point matches to solve the local affine projection. Then the line-to-point distance ratio was calculated to match the lines. In summary, the geometry combined matching tried to obtain the local transformation to match lines with position alignment or enhance the accuracy of line descriptors. It could improve accuracy, while the computation increased exponentially with the number of joint features.

2.3. Matching with multiple view geometry

The matching of two image lines project to a certain position in other images. Thus, when there are more than two images, the geometry constraint for multiple images could be used. Beder et al. (2004) demonstrated that geometric information alone can satisfy the matching task for point and line features. Jain et al. (2010) introduced the global topology across multiple images and used a sweeping approach to avoid explicit line matching. This algorithm could overcome the rupture of line extraction. But it required large computations and might not support large image datasets. To make line matching more efficient, Hofer et al. (2015) used the graphics processing unit (GPU) and the graph cut method to improve line clustering in multiple images. As shown in the study by Li and Yao (2017), Wei et al. (2021a) and Wei et al. (2021b)

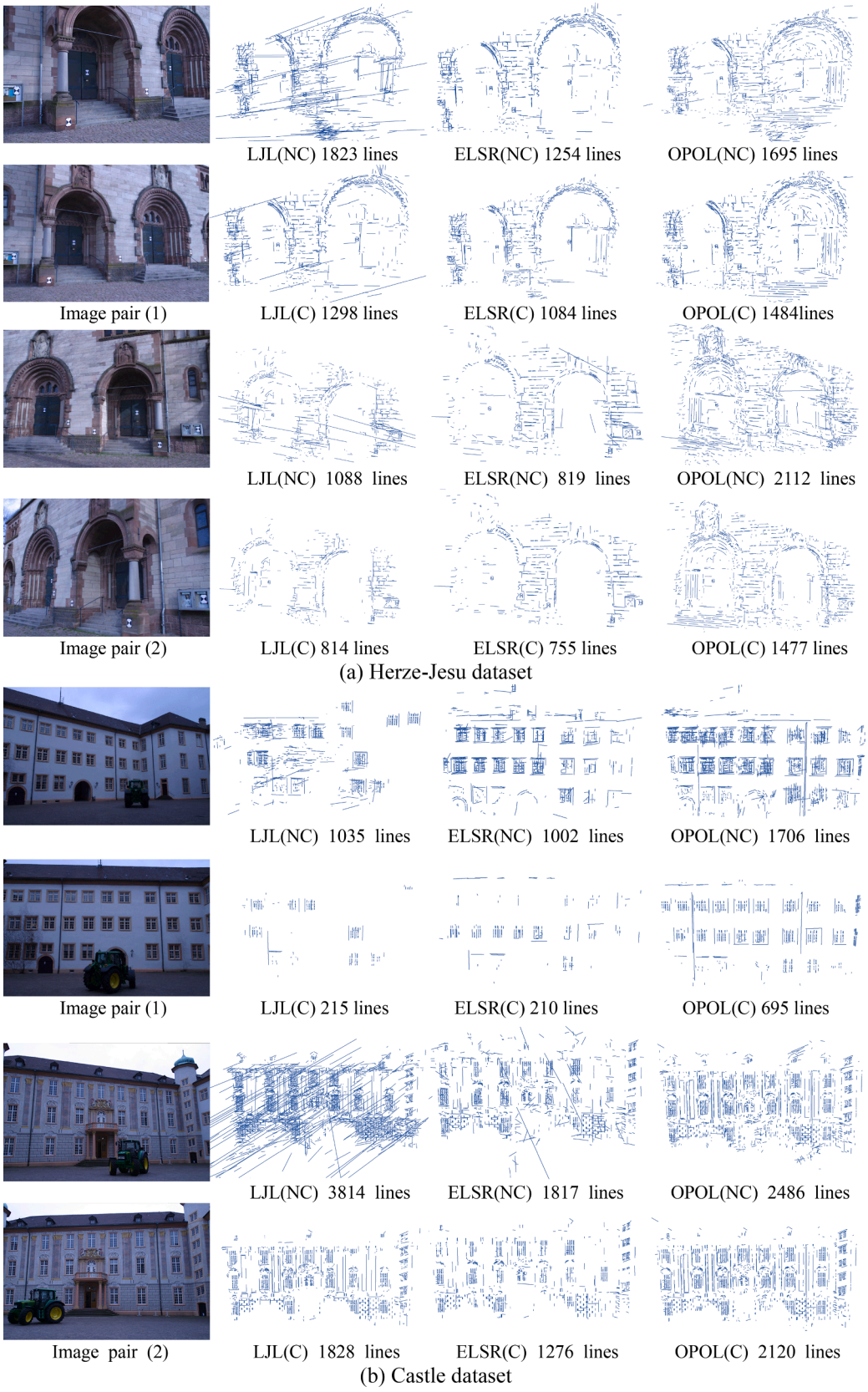


Fig. 6. Visualization of the matching result in ground images.

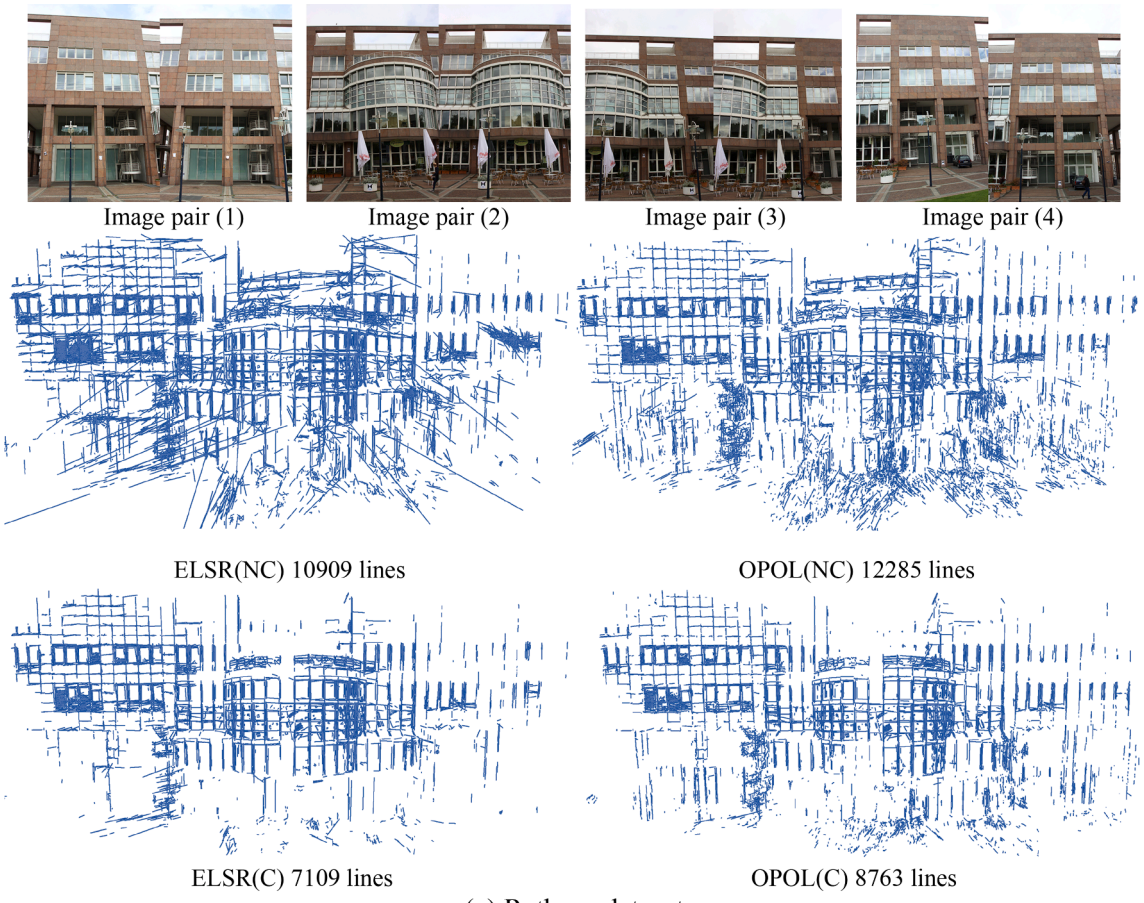


Fig. 6. (*continued*).

Table 2
The summary of matching with ground images.

		Herze-Jesu	Castle	Rathaus
Two-view Task Number		75	87	252
Line Quantity	LJL(NC)	**2028.373**	**2166.56**	-
	ELSR(NC)	1162.76	1407.63	1814.05
	OPOL(NC)	1634.96	1771.09	**2300.39**
Line Quantity With 3D Model	LJL(C)	1199.45	980.23	–
	ELSR(C)	840.64	706.01	1084.52
	OPOL(C)	**1348.16**	**1160.63**	**1487.96**
Match Time(s) on Average	LJL(NC)	488.40	6103.6	–
	ELSR(NC)	1.20	1.17	3.79
	OPOL(NC)	**0.84**	**0.72**	**2.38**
	OPOL(C)	0.85	0.77	2.62

which embedded the alignment of the local homography in a global graph, building the graph for line matching across two or multiple images was helpful to control false positive. But both the graph construction and graph assignment are time-consuming. Thus, Wei et al. (2022) used greedy assignment to find the best line match that satisfied the most homography constraints induced by junction line matches. It significantly improved the speed, while the accuracy was not sacrificed due to the point constraint. Recently, Liu et al. (2023) proposed the LIMAP framework, which improved matching accuracy by vanishing-point constraint, and they used the point match to reduce the degeneracy of line reconstruction.

3. One-point-one-line geometry (OPOL)

The method overview is presented in Fig. 1. Straight line segments and matched points are used as input. OPOL geometry for line correspondence (Section 3.1) contains three main steps:

(1) Given a line segment $\boldsymbol{l}$ in the left image, we construct the sweep trace in the right image on the basis of epipolar geometry and depth constraint. (Section 3.3)
(2) Sweep along the trace in the line map, in which if there is a line segment $\boldsymbol{l}_i^{'}$, OPOL validation is employed with $\boldsymbol{l} \leftrightarrow \boldsymbol{l}_i^{'}$ and the neighbor point matches. The line match with the maximum score is selected as the best match. (Section 3.2)
(3) The endpoint-to-local-mesh distance is used for the final validation if LiDAR point clouds or 3D meshes are available. (Section 3.4)

3.1. The homography of line and point matches

Given a 3D point P and line L, we could obtain the scene plane $\boldsymbol{\pi} = (\boldsymbol{v}^{\top}, 1)^{\top}$, where $\boldsymbol{v} = (v_1, v_2, v_3)^{\top}$ is the normal vector of the plane. For the given $\boldsymbol{\pi}$, there is a projective transformation between the two image planes, and Hartley et al. (2003) showed that this projective transformation is calculated as

$$\mathrm{H} = [\boldsymbol{e}^{'}]_{\times}\mathrm{F} - \boldsymbol{e}^{'}\boldsymbol{v}^{\top} \tag{1}$$

where $\boldsymbol{e}^{'}$ is the epipolar of the right image, $[\boldsymbol{e}^{'}]_{\times}$ is the pseudo inverse matrix of $\boldsymbol{e}^{'}$, and F is the fundamental matrix; these could be induced

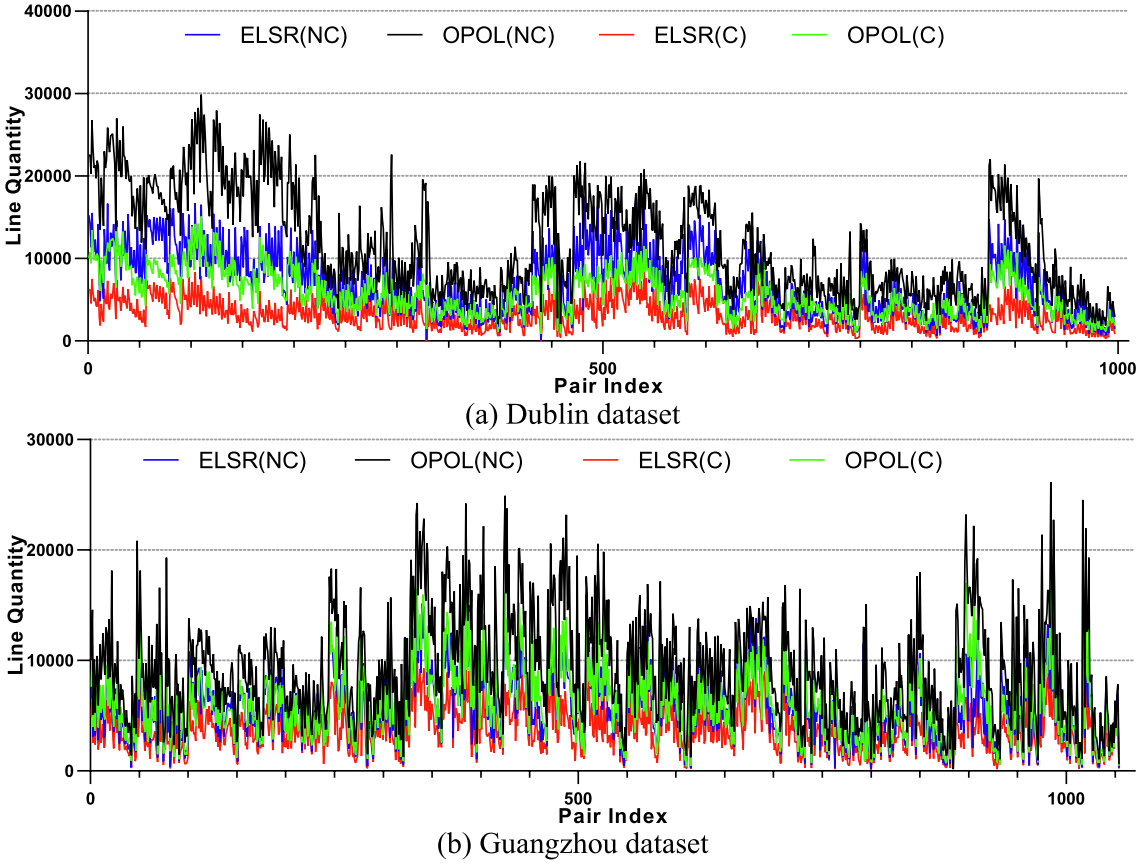


Fig. 7. The statistic of the two-view matching in aerial images.

Table 3
The matching results of aerial images.

		Dublin	Guangzhou
Quantity of image pair		1104	1235
Line quantity	ELSR(NC)	6874.25	5795.78
	ELSR(C)	3124.96	3719.21
	OPOL(NC)	**11748.4**	**9275.20**
	OPOL(C)	5799.18	6166.49
Runtime without line extraction (s)	ELSR(NC)	13.932	10.919
	OPOL(NC)	**1.228**	**1.562**
	OPOL(C)	1.861	4.329

from the camera matrix or the fundament matrix.

Because we have a point and line candidate, which form a plane in 3D space, we can certainly obtain the local projective transformation corresponding to their 3D features. Let $\boldsymbol{l} \leftrightarrow \boldsymbol{l}^{'}$ and $\boldsymbol{p} \leftrightarrow \boldsymbol{p}^{'}$ be the match lines and points in two images. With the H in Equation (1), one could map $\boldsymbol{l}$ and $\boldsymbol{p}$ to the second image as $\boldsymbol{l}^{'}$ and $\boldsymbol{p}^{'}$, respectively. Conversely, if $\boldsymbol{l} \leftrightarrow \boldsymbol{l}^{'}$ and $\boldsymbol{p} \leftrightarrow \boldsymbol{p}^{'}$ are available, we could obtain H via listing the three equations to solve $\boldsymbol{v}$:

$$[\boldsymbol{l}^{'} \quad \boldsymbol{l}^{'} \quad \boldsymbol{p}^{'}]^{\top}(\mathrm{Q} - \boldsymbol{e}^{'}[v_1 \quad v_2 \quad v_3]^{\top})[\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \boldsymbol{p}] = 0 \tag{2}$$

where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are the endpoints of $\boldsymbol{l}$ and $\mathrm{Q} = [\boldsymbol{e}^{'}]_{\times}\mathrm{F}$. The three equations in Equation (2) give an exact solution for the three parameters in $\boldsymbol{v}$. Hartley et al. (2003) give a derivation based on Equations (1)–(2) to directly solve the homography with a point and line match.

$$\mathrm{H} = [\boldsymbol{l}^{'}]_{\times}\mathrm{F} + (\boldsymbol{p}^{'} \times \boldsymbol{e}^{'})^{\top}\frac{(\boldsymbol{p}^{'} \times ((\boldsymbol{F}\boldsymbol{x}) \times \boldsymbol{l}^{'}))}{\|\boldsymbol{p}^{'}\boldsymbol{e}^{'}\|^2(\boldsymbol{l}^{\top}\boldsymbol{p})}\boldsymbol{e}^{'}\boldsymbol{l}^{\top} \tag{3}$$

3.2. Local rotation under the homography

Equation (3) only calculates the local homography with a line and point match, while it cannot answer whether the line match is correct or the two features are indeed coplanar. Thus, under the assumption that the point match $\boldsymbol{p} \leftrightarrow \boldsymbol{p}^{'}$ is a true positive, we introduce the point orientation as additional cues for validation (See Fig. 2). If the point orientations satisfy the local projective transformation, which is induced by the line and point matches, we assume that the line match is correct. The details are described as the follows.

Inspired by Barath (2018), we calculate the local rotation for a pair of specific points under the homography. The relationship between the rotation α and an affine transformation A can be described as

$$\mathrm{A} = \begin{bmatrix} \mathrm{a}_1 & \mathrm{a}_2 \\ \mathrm{a}_3 & \mathrm{a}_4 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} s_u & w \\ 0 & s_v \end{bmatrix} \tag{4}$$

In fact, A is the first-order Taylor approximation of the $3D \rightarrow 2D$ projection function (Molnár et al., 2014), namely, A is the first-order approximation of the homography matrix H. Therefore, given a pair of point match m^{p}, of which $\boldsymbol{p} = [x_1 \quad y_1 \quad 1]^{\top}$ and $\boldsymbol{p}^{'} = [x_2 \quad y_2 \quad 1]^{\top}$, the first-order approximation at $\boldsymbol{p} \leftrightarrow \boldsymbol{p}^{'}$ can be calculated with

$$\begin{aligned} \mathrm{a}_1 &= \frac{\partial x_2}{\partial x_1} = \frac{h_1 - h_7 x_2}{s} \quad & \mathrm{a}_2 &= \frac{\partial x_2}{\partial y_1} = \frac{h_2 - h_8 x_2}{s} \\ \mathrm{a}_3 &= \frac{\partial y_2}{\partial x_1} = \frac{h_4 - h_7 y_2}{s} \quad & \mathrm{a}_4 &= \frac{\partial y_2}{\partial y_1} = \frac{h_5 - h_8 y_2}{s} \end{aligned} \tag{5}$$

where h_* is the row major element of H; s is calculated with $s = x_1 h_7 + y_1 h_8 + h_9$. By combining Equation (4) and (5), we have

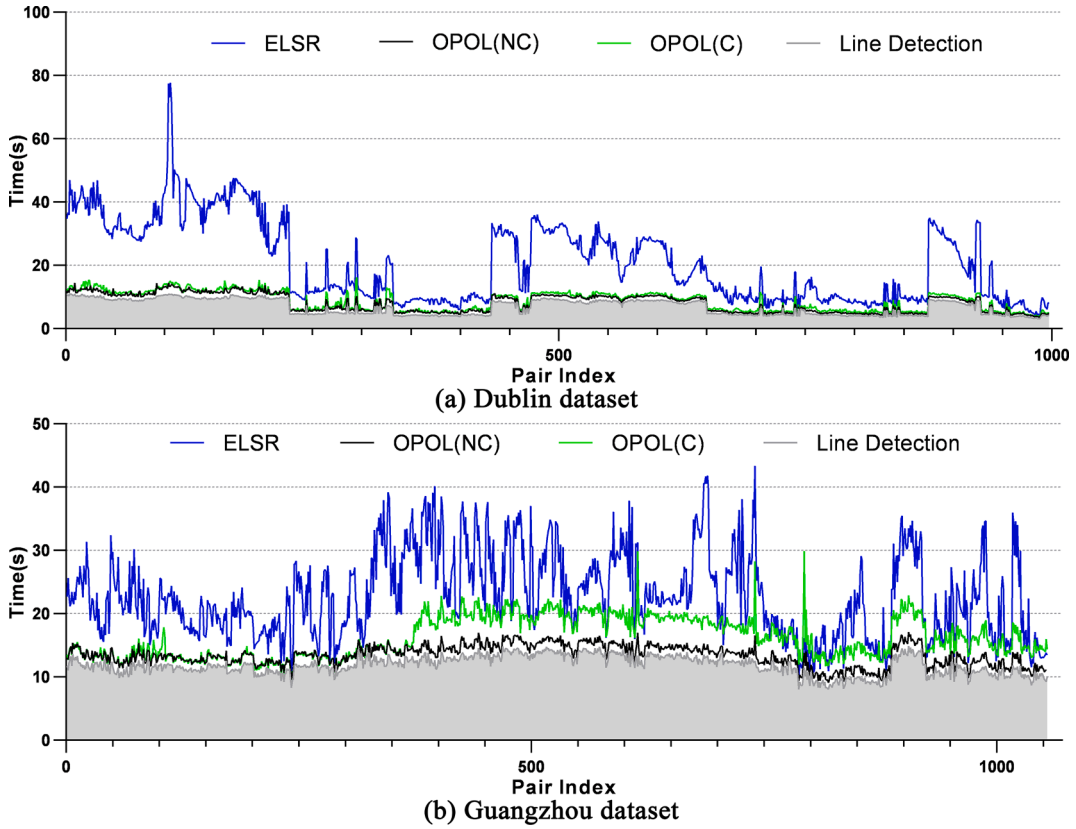


Fig. 8. Runtime in two-view aerial images. The time of the line segment detection is plotted in gray, which takes majority of the proportion in OPOL.

$$\alpha = \arctan\frac{h_4 - h_7 y_2}{h_1 - h_7 x_2} \tag{6}$$

We could score the line match m^{l} by its n neighbor points:

$$s(m^{\mathrm{l}}) = \sum_i \exp\left(-\angle\left(\alpha_i, \angle\left(o_i, o_i^{'}\right)\right)/2t_{ang}\right) \ , \ \text{s.t.} \ \angle\left(o_i, o_i^{'}\right) < t_{ang} \tag{7}$$

where o and $o^{'}$ are the orientations of the left and right point, respectively; $\angle(\bullet)$ calculates the difference of two angles. Note that only when $\angle(o, o^{'})$ is smaller than a giving threshold t_{ang}, the line match will be used for further evaluation.

We can obtain the point orientation (o and $o^{'}$ in Equation (7)) without additional computation. Because the point features extracted by classical point matching methods, such as SIFT (Lowe et al., 2004), SURF (Bay et al., 2008), and other new algorithms (Zhang et al., 2023), are all rotationally invariant. Moreover, the main direction can be extracted directly from the result of these point descriptors.

3.3. Line-sweep matching

Section 3.1 and 3.2 show the validation of the orientation for a point and a line match while in practice we must address the other two contents. First, how to efficiently establish the OPOL candidate, because testing all the candidates would be a time cost. Second, how to resolve contradictory matches when dealing with multiple OPOL candidates. This section shows our line-sweep method to address the two concerns.

As illustrated in Fig. 3, we use the middle point $\boldsymbol{p}_{\mathrm{mid}}$ of line segment $\boldsymbol{l}$ in the left image as the key point to generate the epipolar line in the right image. The line candidate is then searched along the epipolar line. To reduce the sweeping and validation, the depth constraint is used with the simple but reasonable assumption: the line segment is approximately coplanar with its neighbor points. Given the camera matrix and t_{nei} neighbor space points $\{\mathrm{P}_i = (x, y, z, 1)^{\top}\}_{i=1}^{n}$, the depth range can be calculated as

$$\boldsymbol{p}_{\mathrm{mid}} \in [depth_{\min}, depth_{\max}] = \left(\frac{\boldsymbol{m}^{(3)} \cdot \mathrm{P}_i}{\|\boldsymbol{m}^{(3)}\|}\right)_{\min,\max} \qquad i \in [1, n] \tag{8}$$

where $\boldsymbol{m}^{(3)}$ is the third-row of the camera matrix. With $depth_{\min}$ and $depth_{\max}$, the two space points starting from the camera center $\boldsymbol{c}$ and passing through $\boldsymbol{p}_{\mathrm{mid}}$ are

$$\boldsymbol{p}_{[\min,\max]} = \boldsymbol{c} + \frac{[depth_{\min}, depth_{\max}]}{\mathrm{V}_{\mathrm{axi}}^{\top} \cdot \mathrm{V}_{\mathrm{c\ -\ mid}}} \cdot \mathrm{V}_{\mathrm{c\ -\ mid}} \tag{9}$$

where $\mathrm{V}_{\mathrm{c\ -\ mid}}$ is the normalized vector passing through $\boldsymbol{c}$ and $\boldsymbol{p}_{\mathrm{mid}}$; $\mathrm{V}_{\mathrm{axi}}$ is the normalized principal vector. Finally, we could obtain the two image points on the epipolar line as

$$\boldsymbol{p}_{[\min,\max]} = \mathrm{C}^{'} \cdot \left[\mathrm{P}_{[\min,\max]}^{(3)\top}, 1\right]^{\top} \tag{10}$$

which determines the sweeping interval. Finally, any line segment that intersects the epipolar line should be validated with OPOL geometry.

Every line match candidate will have the matching score with Equation (7). If two matches are contradictory, the one with the higher $s(m^{\mathrm{l}})$ will be retained. This resolving process is achieved by the greedy algorithm.

3.4. Space points validation

In practical applications, the dense matching points or LiDAR points may be available. Naturally, we could use them as the ground truth to control OPOL results. Also, they can be used to check whether the result of OPOL is true. With the 3D mesh constructed from the space points, the endpoint-to-mesh distance can be calculated for the 3D line triangulated from the line candidate. Then, the line segment with an abnormal endpoint-to-mesh distance should be incorrect.

Fig. 9. The visualization of matching with aerial images.

Table 4
The statistic of line number and run time in multiple view reconstruction.

Dataset	Method	Time(s)	Line Quantity
Herze-Jesu	Line3D++	12.859	1869
	ELSR	**3.069**	2596
	OPOL	3.227	**2950**
Castle	Line3D++	13.421	2986
	ELSR	3.878	4620
	OPOL	**3.398**	**4847**
Rathaus	Line3D++	51.302	4604
	ELSR	25.628	5801
	OPOL	**21.413**	**7330**
Dublin	Line3D++	831.565	29,030
	ELSR	350.626	251,677
	OPOL	**201.089**	**286,563**
Guangzhou	Line3D++	876.186	26,727
	ELSR	277.843	268,247
	OPOL	**194.36**	**285,652**

Now the key falls into finding the reasonable threshold t_{thr} as the maximum endpoint-to-mesh distance. We follow (Hofer et al., 2015) to calculate t_{thr} by shifting the image pixels and calculating the related shift in object space:

$$t_{\text{thr}} = depth \cdot \tan\left(\angle\left(\mathrm{V}_{\text{axi}}, \overrightarrow{\boldsymbol{cp}_{\text{shift}}}\right)\right) \tag{11}$$

where *depth* is the endpoint depth; $\boldsymbol{c}$ denotes the camera center; V_{axi} is the normalized principal vector as in Equation (9); $\boldsymbol{p}_{\text{shift}}$ is the point shifted from the principal point in the vertical and horizontal direction by σ pixels. We use $\sigma = 1.5$ pixels in validation. Note that the ready-made 3D mesh can be used directly in this validation.

4. Complexity

The time complexity of the algorithm is

$$f(n) = 3kmn + 3mn + km + 3\lambda mn + 3n + m + 1 \tag{12}$$

where n is the number of all line segments in the left image; m denotes

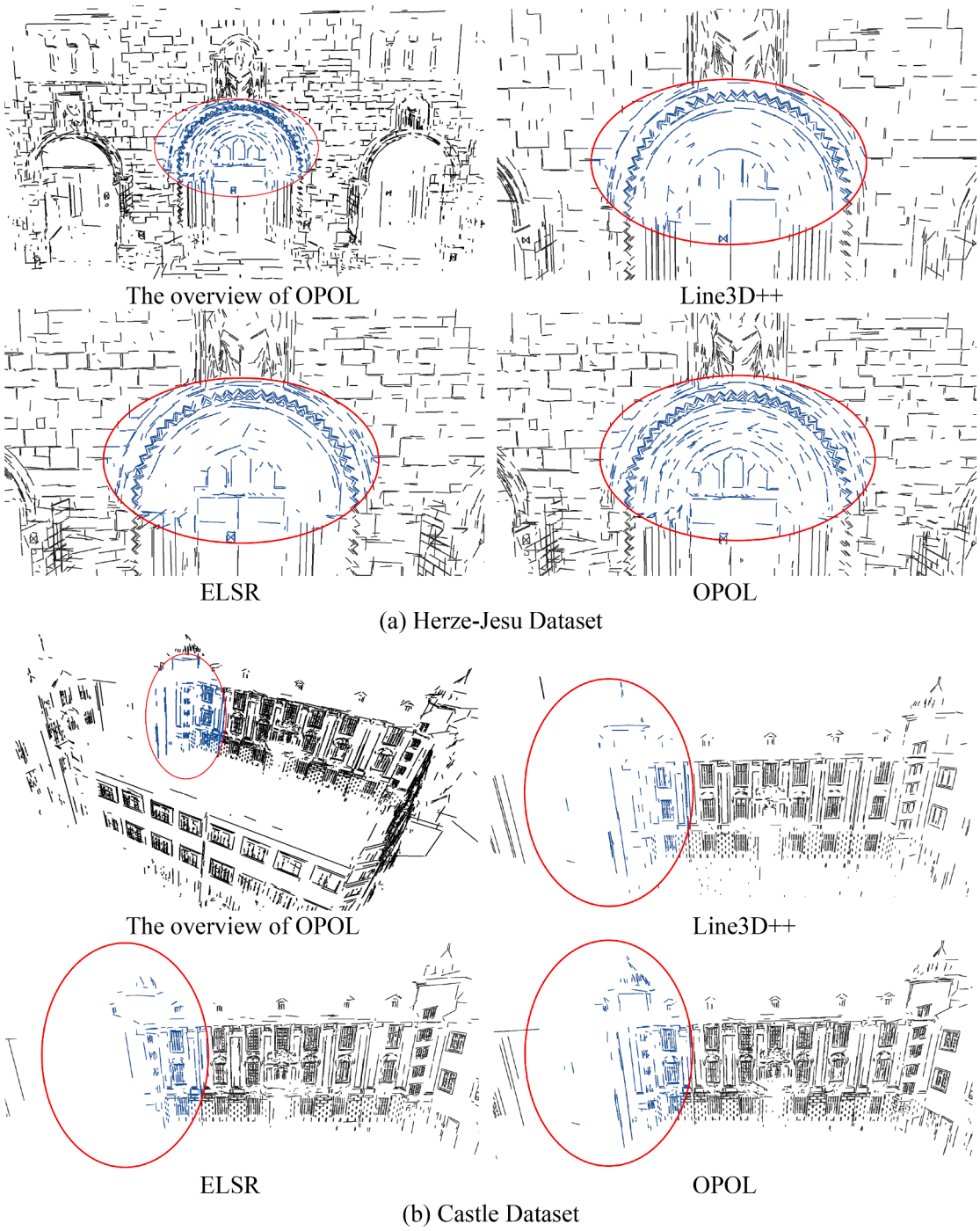


Fig. 10. Results of the 3D line reconstruction. In each dataset, the first image shows an overview of OPOL line reconstruction. The next three images represent the result in local areas. Ellipse highlights the differences.

the number of candidate lines in the right image; $\lambda \in \{0, 1\}$ indicates that there is a probability that no calculation is performed. The coefficient 3 is the sub-iteration inside the main iteration. k is the number of neighbor points around the line segment and is fixed in OPOL ($k = 15$). After omitting the lower-power term of Equation (12), the time complexity is $T(n) = O(mn)$. Because the line sweep strategy, n is much lower than m. Thus, the complexity of this algorithm is bigger than $o(n)$ and is $o(n^2)$ at most. In Section 5, we will further test the speed of OPOL.

5. Experiments and discussions

In this section, we tested OPOL on ground and aerial datasets with high-resolution images, of which the details are presented in Table 1. We preprocessed these images with VisualSFM(Wu et al., 2013) to obtain the camera matrix and tie-points. Because VisualSFM used the SIFT algorithm for point matching, the main directions of the connection points were used directly for OPOL. We compared OPOL with LPI (Fan et al., 2010), LJL (Li and Yao, 2017), GLM (Wei et al., 2021a) and ELSR (Wei et al., 2022) in two-view matching, and we compared OPOL to Line3D++ (Hofer et al., 2015) and ELSR in multiple view line reconstruction.

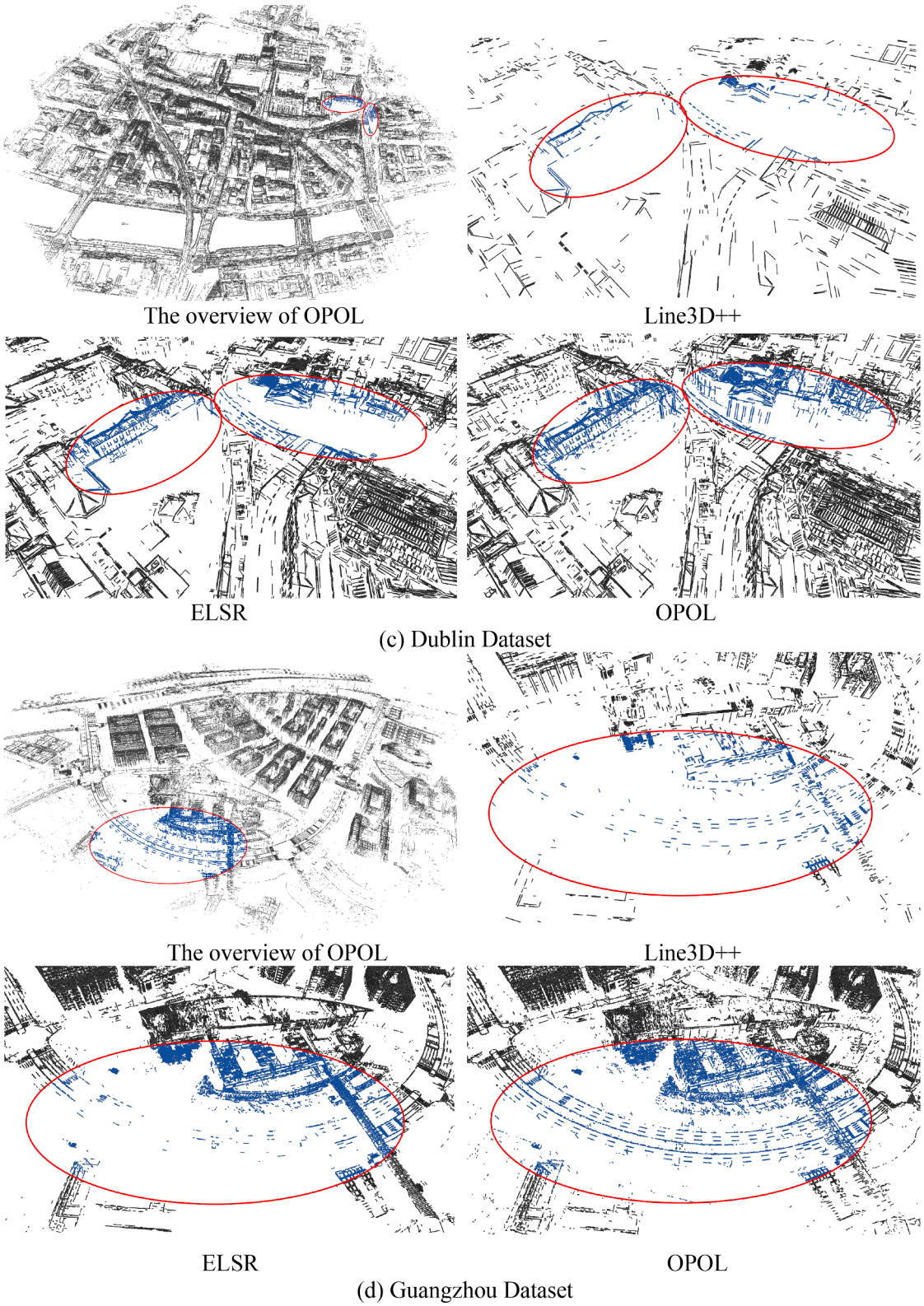


Fig. 10. (*continued*).

We used LSD (Von Gioi et al., 2012) to detect straight line segments for all the methods. In Section 5.5, we also tested OPOL with the line segment from DeepLSD (Pautrat et al., 2023) and the point from FAST (Rosten et al., 2006). For efficiency, our algorithms were implemented in C++ and were tested on a Windows system with an i9 12900Ks CPU and 32 GB of memory.

5.1. Two-view ground images

Three datasets with high-resolution ground image were used for the evaluation of two-view matching. Visual SFM was used to obtain the camera matrix, image pair, and connection points for all algorithms. LPI only requires the connection point, and LJL does not need prior

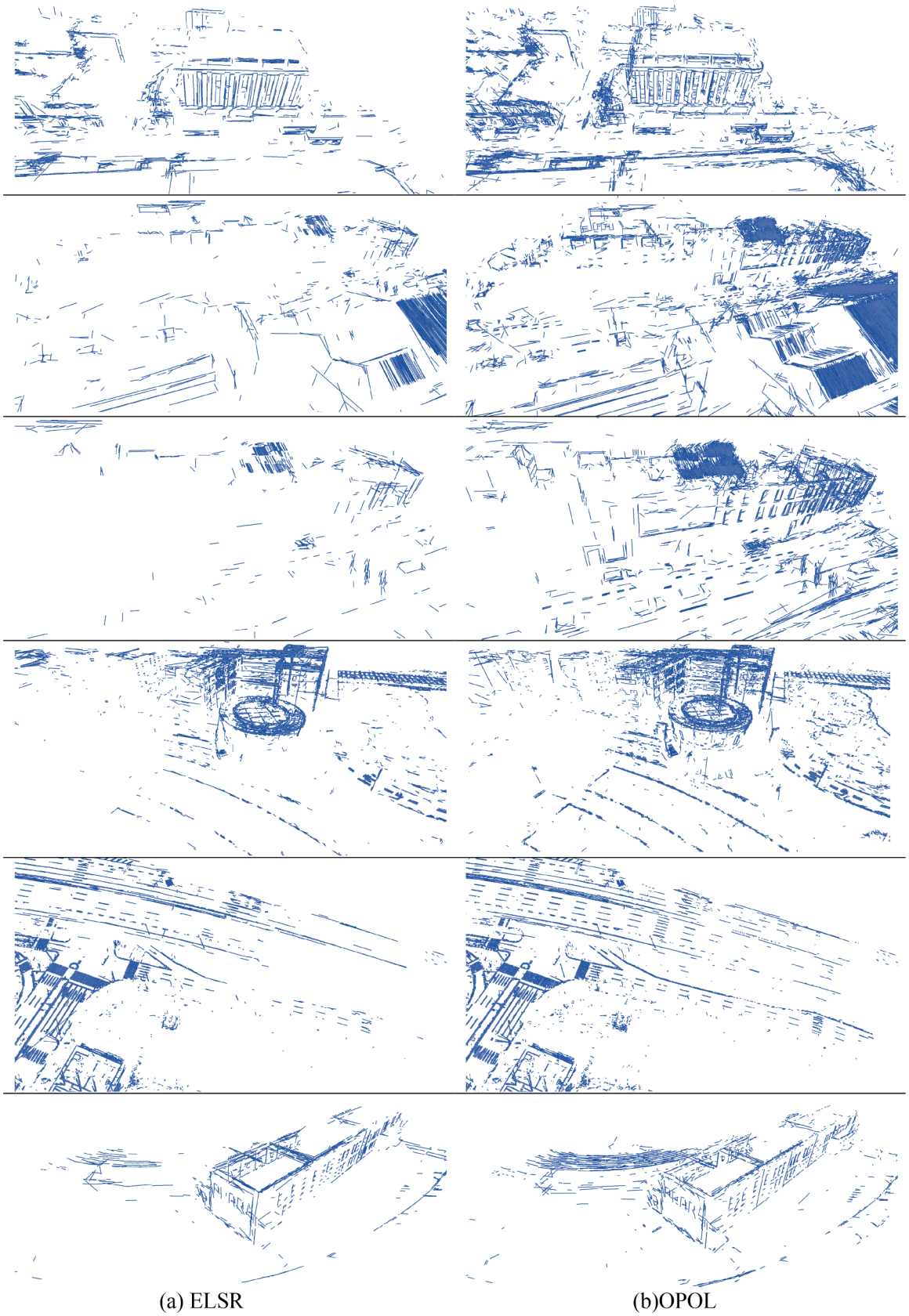


Fig. 11. The six local regions of the two-view matching results.

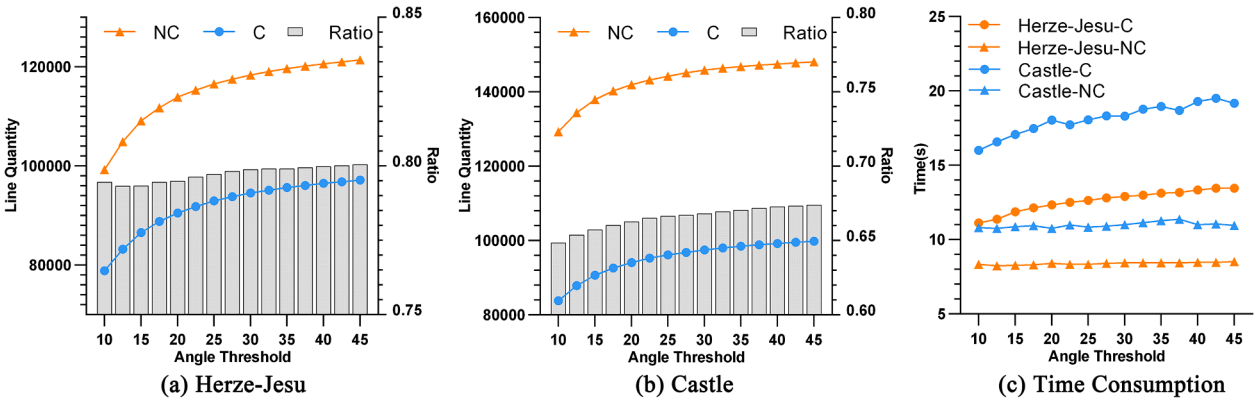


Fig. 12. The influence of altering t_{ang} of OPOL in Herze-Jesu and Castle dataset.

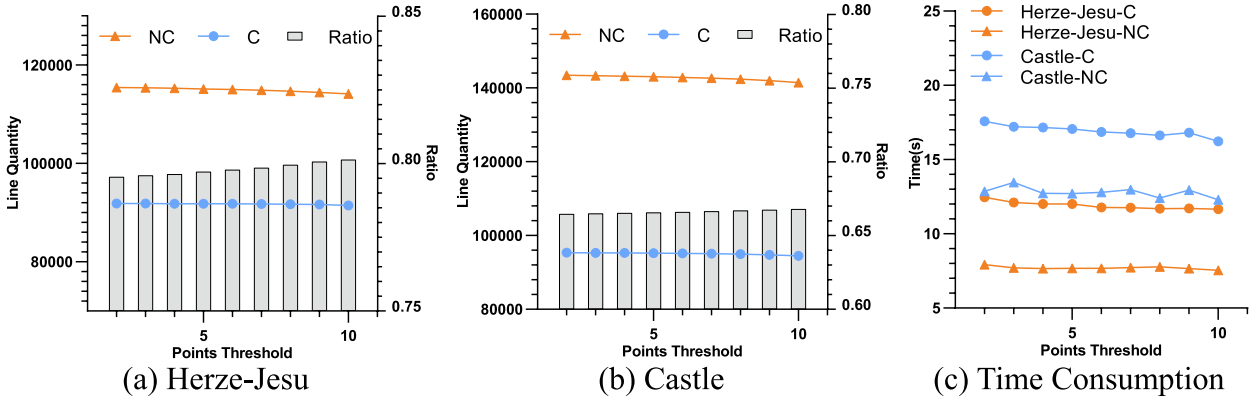


Fig. 13. The influence of altering t_{nei} of OPOL in Herze-Jesu and Castle dataset.

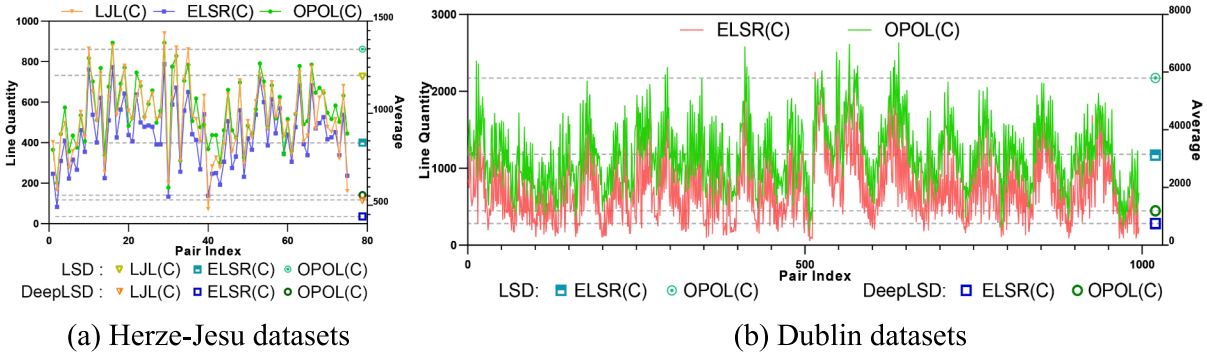
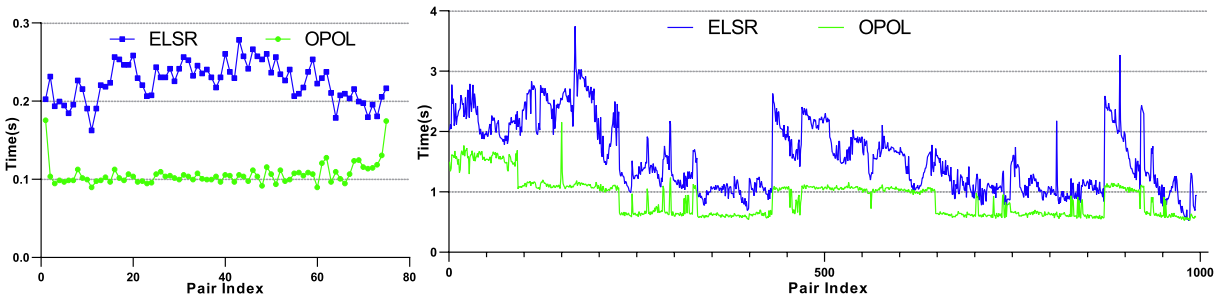


Fig. 14. The correct line matches of matching with the line segment of DeepLSD. The right y-axis draws the average value.

information. We evaluated the two-view matching results with dense point clouds, with which we calculated the endpoint-to-mesh distance and counted the inliers with a specific threshold (the details are presented in Section 3.4).

Fig. 4 and Fig. 5 show the results of LJL, ELSR, and OPOL on the Herze-Jesu, Castle, and Rathaus dataset. For the number of initial matches, the three methods performed similarly in different dataset. LJL obtained the most initial matches for all the datasets, and OPOL acquired more matches than ELSR. After automatically controlling with space points, OPOL produced the most correct line matches, and ELSR was ranked third for correct line matches. Thus, OPOL achieved balanced performance in both quantity and accuracy of line matching, which demonstrated that the proposed one-line-one-point geometry could work. Fig. 6 shows the 3D line segments reconstructed from some two-view matching result. The visualization performs consistently with the statistic: LJL produced more incorrect lines, while OPOL could reconstruct the most correct 3D lines.

Fig. 5 presents the time consumption. Because LJL ran for over 8 min on a pair of images in the dataset, which is far less efficient than ELSR and OPOL, we did not plot it in the graph. We can see that OPOL process the high-resolution image within a second, which was faster than ELSR in most image pairs. The time-consuming gap between ELSR and OPOL increased with the size of the image. In the Rathaus dataset, OPOL was nearly two times faster than ELSR. Statistical data are shown in Table 2. The notation (NC) means that no space points were used for the control. The notation (C) denotes the use of space points for control. Also, the figure showed that the control was quite efficient and took just a little time. The speed advantage is rooted in two aspects. First, ELSR requires



(a) Herze-Jesu datasets (b) Dublin datasets

Fig. 15. The run-time of matching with the DeepLSD line segment.

Table 5
The line quantity of line reconstruction from multiple images.

Dataset	Method	Line Quantity
Herze-Jesu	Line3D++	1237
	ELSR	1722
	OPOL(SIFT)	**1937**
	OPOL(FAST)	1814
Dublin	Line3D++	18,080
	ELSR	101,367
	OPOL(SIFT)	**122,018**
	OPOL(FAST)	97,427

the grouping process for neighbor lines, in which the computation increases with the quantity of the line. Second, the validation geometry of OPOL is simpler and requires less computation than ELSR.

5.2. Two-view aerial images for large areas

To further explore the performance of OPOL, we used two larger datasets of areal images. All two datasets were captured by unmanned aerial vehicles in city scenes. The Guangzhou dataset contains 419 images with the size of 10336 × 7788, and the Dublin dataset contains 369 images with the size of 7360 × 4912. After the structure from motion, we extracted 1104 and 1235 image pairs from the two datasets. Similarly to the evaluation for ground images, we used the dense points and calculated the point-to-mesh distance to verify the correct match.

As shown in Fig. 7, both of the two datasets showed the same trend in the matching results. In large size images, OPOL generally obtained more 3D lines, and OPOL had obvious advantages in the quantity of correct lines after points control. The quantity of lines produced by OPOL and ELSR is presented in Table 3 for the Dublin dataset. OPOL and ELSR obtained 11,748 and 6874 initial line matches on average, respectively. After the control with dense points, OPOL retained nearly 50 % line matches while ELSR retained 45 %, which demonstrated that OPOL was more accurate and produced more matches than ELSR. The performance on the Guangzhou dataset was aligned with the Dublin dataset. OPOL obtained nearly twice the line quantity and achieved better accuracy under point control. Fig. 9 plots some matching results that supported the quantitative analysis well: OPOL could match more small line segments reflecting the details of the structure.

We plot the run-time of the two methods in Fig. 8, and we present the summary in Table 3. There are three remarkable features for the run-time. First, OPOL showed the great advantage in speed when dealing with large-size images; it was more than ten times faster than ELSR because ELSR required large computations to group and validate the junction line; also, the runtime of ELSR raised quickly with the number of single lines and junction planes. Second, the matching time of OPOL was closely related to the line detection time, which indicates that OPOL runs in a linear time since it was only influenced by the line quantity. Third, in comparison to the time consumption of the line segment matching portion, the time taken for line segment extraction constitutes the majority of the program's execution time; thus, a fast line detection algorithm would be quite helpful to speed up the whole algorithm.

5.3. Line reconstruction of multiple images

The experiments in two-view matching showed that OPOL could obtain more matches and significantly improved the time efficiency compared to other algorithms. In this section, we further explore how the two-view matching algorithm contributes to the performance of the line reconstruction of multiple images. We directly embedded OPOL into the ELSR framework to replace its two-view matching algorithm. Except for this, OPOL and ELSR shared the same inputs and algorithms. Line3D++, the robust line reconstruction algorithm is evaluated for comparison.

Table 4 presents the reconstruction result. Line3D++ is the slowest in speed and has reconstructed the least 3D lines. OPOL performed similarly at runtime on Herze-Jesu and Castle datasets compared to ELSR. The run-time was significantly improved in the two large datasets (Guangzhou and Dublin). For the line quantity, OPOL still performed better than ELSR. This result was expected because of the better performance of OPOL in two-view. As shown in Fig. 10, OPOL reconstructed many detailed lines from multiple images.

Note that the reconstruction result could not fully reflect the advantage of OPOL. Because the line merging constraints in ELSR might wrongly eliminate many correct matches to highlight the key 3D line segments. We could have an intuitive look at all the two-view matching results in Fig. 11, in which both the global picture and the subfigure showed that OPOL produced more correct correspondences. Thus, OPOL has great potential to be a robust and effective unit in efficient and accurate line segment reconstruction.

5.4. The hyperparameters

The OPOL geometry has an important hyperparameter t_{ang} (Equation (7)), which determines whether one point can be added to form the OPOL score of the line candidate. In this section, we altered t_{ang} from 10 to 45 degrees to explore its influence in Herze-Jesu and Castle dataset. As shown in Fig. 12, we counted the time and line quantity in different t_{ang}. OPOL performed consistently in the two datasets and we could conclude their common features. First, t_{ang} was closely related to the line quantity. A bigger t_{ang} would produce more line matches, especially for the wide-baseline pair. It lowered the constraint to be a valid match. Second, t_{ang} had little influence on the accuracy. As shown in the figure, the results of both OPOL(NC) and OPOL(C) raised with the increase of t_{ang}, and the accuracy barely changed. Third, the runtime remained steady in different t_{ang}. As discussed in the first feature, although a bigger t_{ang} would bring about more validations in the line match, the runtime just increased a little, which also demonstrated the low complexity of the OPOL geometry validation.

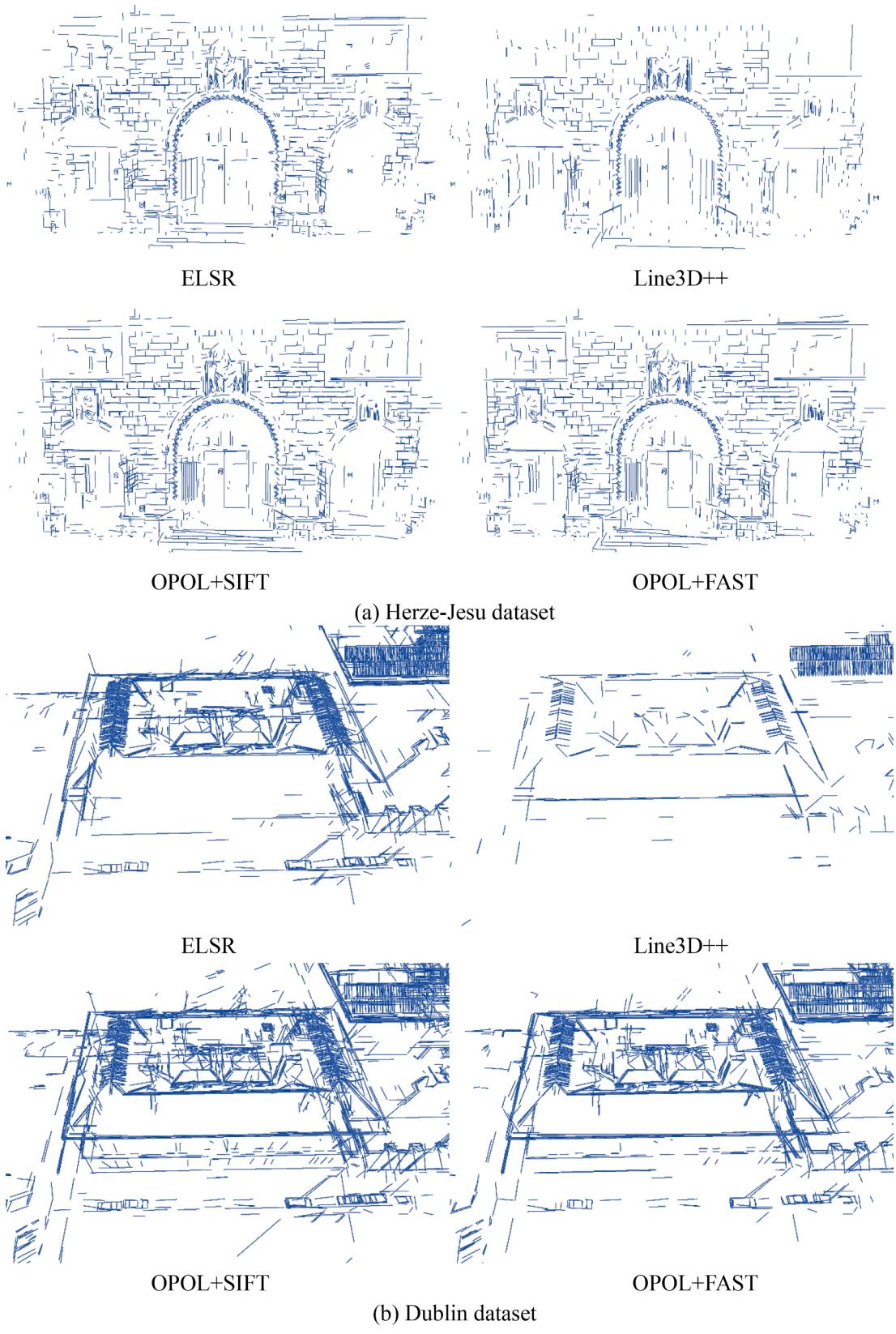


Fig. 16. The local result of the line reconstruction with different features.

In Section 5, we also used the hyperparameter t_{nei} to give the false positive control, which constrained the least number of neighbors for the line candidate to satisfy Equation (7). We altered t_{nei} from 2 to 10 to evaluate the performance of OPOL. As plotted in Fig. 13, with the increase of t_{nei}, the line quantity slightly decreased and the slope of the change was obviously smaller than t_{ang}, which demonstrated that $t_{nei} = 4$ in our paper was robust enough. As for the runtime, we could see that it decreased with the increase of t_{nei}. Because a bigger t_{nei} will reduce the

initial line candidates and save time in the following process. Also, compared with t_{ang}, changing t_{nei} brought about less influence on overall performance.

5.5. Matching with different line and point features

To evaluate the performance of OPOL with different inputs, we used DeepLSD (Pautrat et al., 2023) and FAST (Rosten et al., 2006) as the line and point feature to replace LSD and SIFT in the above experiments. DeepLSD is the state-of-the-art line detector with deep learning strategy, and the FAST algorithm obtains the feature point with orientations efficiently. We reduced the image size in the Dublin dataset by 1/3 to make DeepLSD executable on NVIDIA RTX 3090. Both the validation strategy and the hyperparameters were the same as those of the previous experiments.

Fig. 14 and Fig. 15 show the quantity and the run time for all image pairs, both of which decreased because DeepLSD extracted fewer line segments and the line quantity was reduced. Compared to other line matching methods, we can draw the same conclusion as in Section 5.1: OPOL obtained more correct matches in less time in these datasets. The results of two-view matching directly influenced the reconstruction with multiple images. As shown in Table 5, all the methods reconstructed fewer 3D lines. However, Fig. 16 shows a better visualization because of the improvement in 2D line detection with DeepLSD. Also, the feature point influenced the result. In Table 5, OPOL produced fewer line segments when using the FAST point, mainly because it is less accurate than SIFT in both position and rotation. These experiments also indicate that line reconstruction for multiple images is a comprehensive work, in which the result of the sub-step will influence the final reconstruction.

6. Conclusion

In this paper, we propose the novel one-point-one-line geometry (OPOL) for two view line segment correspondence, which verifies the line match with only one point match by exploiting the point orientation. Due to the simple grouping and validation process of OPOL, efficient experiments with thousands of image pairs showed that it ran more efficiently and generally produced more matches than previous algorithms.

There are two drawbacks of the proposed method. First, OPOL is highly dependent on point matching, which limits its performance when point matching is not robust. Also, OPOL would fail in the weak texture scene, where point matching cannot work. Second, OPOL only works when the line correspondences are on the same plane with the point matches. Thus, the performance may be unsatisfied in a scene with many manifold shapes.

There are two significant works in the future. First, the accurate line merging algorithm in multiple images should be explored, which could fully make use of the two-view matching result. Second, two-view line reconstruction has no constraint and is easier than point to be degenerate (Ok et al., 2012b); since we have established the coplanar geometry with one point and one line, the point orientation could be used to improve the accuracy of line reconstruction from both two and multiple images.

CRediT authorship contribution statement

Haoyu Guo: Conceptualization, Data curation, Methodology, Software, Writing – original draft, Writing – review & editing. **Dong Wei:** Conceptualization, Methodology, Writing – review & editing, Funding acquisition. **Yongjun Zhang:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Yi Wan:** Funding acquisition, Resources. **Zhi Zheng:** Validation, Visualization. **Yongxiang Yao:** Validation, Visualization. **Xinyi Liu:** Funding acquisition, Visualization. **Zhuofan Li:** Investigation, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the fellowship of National Natural Science Foundation of China under Grant 42001406, China National Postdoctoral Program for Innovative Talents under Grant Bx2022024, National Natural Science Foundation of China under Grant 42201474.

References

Abdellali, H., Frohlich, R., Vilagos, V., Kato, Z., 2021. L2d2: learnable line detector and descriptor. 2021 International Conference on 3D Vision (3DV), IEEE.
Al-Shahri, M., Yilmaz, A., 2014. Line matching in wide-baseline stereo: a top-down approach. IEEE Trans. Image Process. 23 (9), 4199–4210.
Barath, D., 2018. Five-point fundamental matrix estimation for uncalibrated cameras. CVPR.
Bay, H., Ferraris, V., Van Gool, L., 2005. Wide-baseline stereo matching with line segments. CVPR.
Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (SURF). Comput. Vis. Image Underst. 110 (3), 346–359.
Beder, C. (2004). A unified framework for the automatic matching of points and lines in multiple oriented images. Proc. 20th ISPRS Congress, Istanbul, Turkey, Citeseer.
Chen, M., Yan, S., Qin, R., Zhao, X., Fang, T., Zhu, Q., Ge, X., 2021. Hierarchical line segment matching for wide-baseline images via exploiting viewpoint robust local structure and geometric constraints. ISPRS J. Photogramm. Remote Sens. 181, 48–66.
Fan, B., F. Wu and Z. Hu (2010). Line matching leveraged by point correspondences. 2010 CVPR.
Gong, Y., Zhou, P., Liu, Y., Dong, H., Li, L., Yao, J., 2023. View-graph key-subset extraction for efficient and robust structure from motion. Photogram. Rec. 00, 1–45.
Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision. Cambridge University Press.
Hofer, M., M. Maurer and H. Bischof (2015). Line3d: Efficient 3d scene abstraction for the built environment. Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings 37, Springer.
Jain, A., Kurz, C., Thormählen, T., Seidel, H.-P., 2010. Exploiting global connectivity constraints for reconstruction of 3D line segments from images. CVPR.
Jia, X., Huang, X., Zhang, F., Gao, Y., Yang, C., 2019. Robust line matching for image sequences based on point correspondences and line mapping. IEEE Access 7, 39879–39896.
Lange, M., C. Raisch and A. Schilling (2020). WLD: A wavelet and learning based line descriptor for line feature matching. Vision, Modeling, and Visualization, The Eurographics Association.
Lange, M., Schweinfurth, F., Schilling, A., 2019. Dld: a deep learning based line descriptor for line feature matching. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE.
Li, K., Yao, J., 2017. Line segment matching and reconstruction via exploiting coplanar cues. ISPRS J. Photogramm. Remote Sens. 125, 33–49.
Li, K., Yao, J., Lu, X., Li, L., Zhang, Z., 2016. Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation. Neurocomputing 184, 207–220.
Liu, S., Yu, Y., Pautrat, R., Pollefeys, M., Larsson, V., 2023. 3D line mapping revisited. CVPR.
López, J., Santos, R., Fdez-Vidal, X.R., Pardo, X.M., 2015. Two-view line matching algorithm based on context and appearance in low-textured images. Pattern Recogn. 48 (7), 2164–2184.
Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60, 91–110.
Molnár, J., Chetverikov, D., 2014. Quadratic transformation for planar mapping of implicit surfaces. J. Math. Imaging Vision 48, 176–184.
Ok, A.O., Wegner, J.D., Heipke, C., Rottensteiner, F., Soergel, U., Toprak, V., 2012a. Matching of straight line segments from aerial stereo images of urban areas. ISPRS J. Photogramm. Remote Sens. 74, 133–152.
Ok, A.Ö., Wegner, J.D., Heipke, C., Rottensteiner, F., Sörgel, U., Toprak, V., 2012b. Accurate reconstruction of near-epipolar line segments from stereo aerial images. Photogramm. Fernerkund. Geoinf 2012, 345–358.
Pautrat, R., et al., 2023. Deeplsd: line segment detection and refinement with deep image gradients. CVPR.
Pautrat, R., J.-T. Lin, V. Larsson, M. R. Oswald and M. Pollefeys (2021). SOLD2: Self-supervised occlusion-aware line description and detection. CVPR.
Ramalingam, S., Antunes, M., Snow, D., Hee Lee, G., Pillai, S., 2015. Line-sweep: cross-ratio for wide-baseline matching and 3d reconstruction. CVPR.
Rosten, Edward, and Tom Drummond. Machine learning for high-speed corner detection. 2006 Computer Vision–ECCV.
Schmid, C., Zisserman, A., 1997. Automatic line matching across views. CVPR.
Shi, J., Wang, X., 2017. A local feature with multiple line descriptors and its speeded-up matching algorithm. Comput. Vis. Image Underst. 162, 57–70.

Sun, Y., Zhao, L., Huang, S., Yan, L., Dissanayake, G., 2015. Line matching based on planar homography for stereo aerial images. ISPRS J. Photogramm. Remote Sens. 104, 1–17.
Tola, E., Lepetit, V., Fua, P., 2009. Daisy: an efficient dense descriptor applied to wide-baseline stereo. IEEE Trans. Pattern Anal. Mach. Intell. 32 (5), 815–830.
Vakhitov, A., Lempitsky, V., 2019. Learnable line segment descriptor for visual slam. IEEE Access 7, 39923–39934.
Von Gioi, R.G., Jakubowicz, J., Morel, J.-M., Randall, G., 2012. LSD: a line segment detector. Image Process. Line 2, 35–55.
Wang, Z., Wu, F., Hu, Z., 2009. MSLD: a robust descriptor for line matching. Pattern Recogn. 42 (5), 941–953.
Wang, Q., Zhang, W., Liu, X., Zhang, Z., Baig, M.H.A., Wang, G., He, L., Cui, T., 2020. Line matching of wide baseline images in an affine projection space. Int. J. Remote Sens. 41 (2), 632–654.
Wang, J., Zhu, Q., Liu, S., Wang, W., 2021. Robust line feature matching based on pair-wise geometric constraints and matching redundancy. ISPRS J. Photogramm. Remote Sens. 172, 41–58.
Wei, D., Y. Wan, Y. Zhang, X. Liu, B. Zhang and X. Wang (2022). ELSR: Efficient line segment reconstruction with planes and points guidance. CVPR.
Wei, D., Zhang, Y., Li, C., 2021a. Robust line segment matching via reweighted random walks on the homography graph. Pattern Recogn. 111, 107693.
Wei, D., Zhang, Y., Liu, X., Li, C., Li, Z., 2021b. Robust line segment matching across views via ranking the line-point graph. ISPRS J. Photogramm. Remote Sens. 171, 49–62.
Wu, C., 2013. Towards Linear-Time Incremental Structure from Motion. IEEE.
Zhang, L., Koch, R., 2013. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. J. Vis. Commun. Image Represent. 24 (7), 794–805.
Zhang, Y., Yao, Y., Wan, Y., Liu, W., Yang, W., Zheng, Z., Xiao, R., 2023. Histogram of the orientation of the weighted phase descriptor for multi-modal remote sensing image matching. ISPRS J. Photogramm. Remote Sens. 196, 1–15.
Zhang, C., 2002. Updating of Cartographic Road Databases by Image Analysis, Doctoral Thesis, ETH Zurich, Diss. ETH No. 14934.