

Sensing Elsevier Editorial System(tm) for ISPRS Journal of Photogrammetry and Remote  
Manuscript Draft

Manuscript Number: PHOTO-D-14-00307R3

Title: Line Matching based on Planar Homography for Stereo Aerial Images

Article Type: Original Research Paper

Keywords: Line Matching; Planar Homography; Point Feature Correspondences; Coplanarity; Bundle Adjustment.

Corresponding Author: Mr. Lei Yan,

Corresponding Author's Institution:

First Author: Yanbiao Sun

Order of Authors: Yanbiao Sun; Liang Zhao; Shoudong Huang; Lei Yan; Gamini Dissanayake

## Responses to Reviewers

Paper No. PHOTO-D-14-00307R2  
Title "Line Matching based on Planar Homography for Stereo Aerial Images"  
Author(s) Yanbiao Sun, Liang Zhao, Shoudong Huang, Lei Yan, Gamini Dissanayake

ISPRS Journal of Photogrammetry and Remote Sensing --- Publication with Minor revision

The authors would like to thank the editor and the anonymous reviewer for spending valuable time to review the paper and providing the detailed comments. Now the paper has been revised according to the suggestions and comments. Below are the list of major changes and detailed responses to the comments. For the convenience of the reviewers, we have kept all the original comments in this reply.

### Major changes in the new version:

- (1) Some inaccurate statements are removed or modified.*
- (2) The computation time for BA is now added into the total matching time in Table 4.*
- (3) Section 4 is rewritten to make the idea and process clearer.*
- (4) Reference (Heuel & Foerstner) is added and discussed in Section 1 and Section 7.*

## **Reply to the editor's comments and suggestions:**

Dear Mr. Lei Yan,

Reviewers have now commented on your paper. I am pleased to say that it has been favourably received and publication with Minor revision is recommended (see below and on <http://ees.elsevier.com/photo/>). If you are prepared to undertake the work required, I would be pleased to reconsider the revised paper for publication.

For your guidance, reviewers' comments are appended below.

To submit a revision, please go to <http://ees.elsevier.com/photo/> and login as an Author.

Your username is: \*\*\*\*\*

If you need to retrieve password details, please go to:  
[http://ees.elsevier.com/photo/automail\\_query.asp](http://ees.elsevier.com/photo/automail_query.asp)

NOTE: Upon submitting your revised manuscript, please upload the source files for your article. For additional details regarding acceptable file formats, please refer to the Guide for Authors at: [http://www.elsevier.com/journals/isprs-journal of-photogrammetry and-remote-sensing/0924-2716/guide-for-authors](http://www.elsevier.com/journals/isprs-journal-of-photogrammetry-and-remote-sensing/0924-2716/guide-for-authors)

When submitting your revised paper, we ask that you include the following items:

### **Response to Reviewers (mandatory)**

This should be a separate file labeled "Response to Reviewers" that carefully addresses, point-by-point, the issues raised in the comments appended below. You should also include a suitable rebuttal to any specific request for change that you have not made. Mention the page, paragraph, and line number of any revisions that are made.

### **Manuscript and Figure Source Files (mandatory)**

We cannot accommodate PDF manuscript files for production purposes. We also ask that when submitting your revision you follow the journal formatting guidelines. Figures and tables may be embedded within the source file for the submission as long as they are of sufficient resolution for Production. For any figure that cannot be embedded within the source file (such as \*.PSD Photoshop files), the original figure needs to be uploaded separately. Refer to the Guide for Authors for additional information. [http://www.elsevier.com/journals/isprs-journal of-photogrammetry and-remote-sensing/0924-2716/guide-for-authors](http://www.elsevier.com/journals/isprs-journal-of-photogrammetry-and-remote-sensing/0924-2716/guide-for-authors)

### **Graphical Abstract (optional)**

Graphical Abstracts should summarize the contents of the article in a concise, pictorial form designed to capture the attention of a wide readership online. Refer to the following website for more information: <http://www.elsevier.com/graphicalabstracts>

On your Main Menu page is a folder entitled "Submissions Needing Revision". You will find your submission record there.

Given that the requested revisions are fairly minor the new version is required within 2 weeks.

If you feel that you will be unable to submit your revision within the time allowed, please contact our editorial office to discuss an extension to your revision deadline.

Please proceed to the following link to update your personal classifications and keywords, if necessary:

\*\*\*\*\*

PLEASE NOTE: The journal would like to enrich online articles by visualising and providing geographical details described in ISPRS Journal of Photogrammetry and Remote Sensing articles. For this purpose, corresponding KML (GoogleMaps) files can be uploaded in our online submission system. Submitted KML files will be published with your online article on ScienceDirect. Elsevier will generate maps from the KML files and include them in the online article.

Please note that this journal offers a new, free service called AudioSlides: brief, web-cast-style presentations that are shown next to published articles on ScienceDirect (see also <http://www.elsevier.com/audioslides>). If your paper is accepted for publication, you will automatically receive an invitation to create an AudioSlides presentation.

PLEASE NOTE: ISPRS Journal of Photogrammetry and Remote Sensing would like to enrich its relevant online articles by displaying 3D models that allow the reader to interactively explore the underlying research data. Hence, if applicable, we would like to invite you to upload with your manuscript 3D models (in OBJ, PLY or U3D data format) as supplementary material to our online submission system. Elsevier will generate the interactive viewer for your datasets and include it with the online article on ScienceDirect. More information can be found at:

<http://www.elsevier.com/about/content-innovation/obj-ply-models> and

<http://www.elsevier.com/about/content-innovation/u3d-models>

Yours sincerely,

Konrad Schindler, PhD

Associate Editor

ISPRS Journal of Photogrammetry and Remote Sensing

## Reviewers' comments:

The manuscript has improved, nevertheless a further minor revision is required. Please address the points listed below.

Technical Issues:

-----  
(p719ff) I don't understand the argument "we only know projection matrices". Please elaborate. It is trivially possible to decompose a projection matrix into its components  $K, R, t$ . So what are you trying to say? On the same page, the derivation is obvious, equations (3) and (4) are not needed.

*Reply: You are right. Thank you for pointing out this. We have now modified this sentence like: "For some cases (e.g. the publicly available Aerial\_Large dataset (Werner, 2007)), only two projection matrices are provided. To compute homography in these cases, one way is to first decompose the projection matrix into calibration matrix, camera rotational matrix and translation, another way is to use the projection matrices directly. The formula of computing homography using two camera projection matrices is deduced below." (See Page 7, Paragraph 1, Line 9~16)*

*Equations (3) and (4) are deleted as suggested.*

(p9131ff) Please make the statement (only 3D points on one side are coplanar) more precise. Not every line is a crease edge; it can just be a texture edge (like a road marking on a flat road). Then the 3D points on both sides of the line segment *\*are\** coplanar.

*Reply: We have modified this sentence like: "In addition, because we use the 3D points on each side of the line to fit one plane, we divide the 3D points into two sets." (See Page, Paragraph, Line)*

(p10151ff) Again, I do not understand your argumentation regarding estimation uncertainty. You can do error propagation to get the uncertainty of the line itself (not its endpoints), and can then determine whether two lines coincide or not, whether a point lies on the line or not etc. by statistical tests. See for example [Heuel & Foerstner, CVPR'01]. So why a threshold and not uncertainty propagation? The same thing applies on (p1219). Why not use the uncertainty and do a proper statistical test to determine whether a line segment match is valid?

*Reply: We think the method of using uncertainty of the predicted line and applying statistical tests is very interesting and will consider it in our future work.*

*In this paper, we obtain a predicted line using Equation (8).*

$$l_E \propto (K^{-T} R K^T + \frac{K^{-T} R n_L t^T R K^T}{d - n_L^T R^T t})(x_l^2 \times x_l^1)$$

*As the result of BA, the uncertainty of three rotational angles, three translation vec-*

tors and 3D points can be provided. The uncertainty of the fitted plane can be obtained by the uncertainty of the 3D points and the plane fitting formula. To obtain the uncertainty of the predicted line, we need to first calculate the Jacobian with respect to three rotational angles, translation vectors and the parameters of the fitted plane. With the known uncertainty, applying statistical tests similar to method in (Heuel, 2001) can achieve line matching. However,, the process to calculate the Jacobian is a bit complicated due to the complicated formulas used (e.g. Equation (8)) thus this method is not implemented. We have now removed the unsuitable explanation and rephrase the sentences as follows.

*“In this paper, to simplify the algorithm, a line segment will be chosen as a candidate one using a threshold instead of using statistical tests by computing the uncertainty of the line.” (See Page 10, Paragraph 1, Line 33~35)*

*“To make the algorithm simpler, if the orientation and position shift are less than given thresholds, the candidate will be regarded as the line segment corresponding to the original source line segment.”(See Page 11, Paragraph 2, Line 52~54)*

*“In future work, we will investigate how to use the uncertainty of predicted line to choose the tentative line segments rather than using a given threshold.” (See Page 35, Paragraph 2, Line 48~50 )*

*In addition, in introduction section, we added the description about the paper of Heuel & Foerstner (Heuel & Foerstner, 2001) (See Page 2, Paragraph 3, Line 54~56 )*

(p12l30) Again unclear - why is only one of multiple corresponding one segments found? It would seem more natural to allow one-to-many matching or even many-to-many matching, this would additionally allow one to improve the monocular grouping and get longer lines. Please upgrade.

*Reply: In our matching method, the similarity between all tentative line segments and the given line segment on the source image will be calculated and the maximum similarity will be chosen to determine whether the line segment with maximum similarity is matched. Thus, it allows many-to-one matches, similar to the work of Schmid and Zisserman, 1997 and Fan et al., 2012 as following.*

*“Cases where there are multiple matches possible for a particular line segment are resolved by a winner takes all scheme.” (Schmid and Zisserman, 1997)*

*“Given a pair of lines, we calculate their similarity by corresponding points in their neighborhood of right and left side separately. Then the maximum of these two values is taken as the final similarity of these two lines. It allows many-to-one correspondences which happens when a line in one image is split into multi-segments.”(Fan et al., 2012)*

*If we allow the case of one-to-many matches, then some tentative lines will be regarded as matches when their similarities are larger than a given threshold. This operation will result in more matches but also result in more wrong matches.*

*Now we have modified this sentence as “When there are multiple possible matches*

*whose similarities are greater than a given threshold, the method can generate one-to-many matches if only the threshold is applied. To reduce the number of wrong matches, the tentative line with maximum similarity will be regarded as the final match in this paper.” (See Page 11, Paragraph 2, Line 55~56 & Page 12, Paragraph 1, Line 9~12)*

(p13, sec 4) This part is conceptually unclear. First of all, you explain how to get a homograph for a parallel plane (sec 4.1), but then go on to use the terrain plane instead (sec 4.2). How are these two now combined? Do you always use the terrain plane? Or do you use planes that are parallel to the terrain, but shifted along the vertical? Then why not directly use the fact that in any aerial imaging project the object-space vertical is known? Also, if the method with the terrain plane is accurate enough to correctly match lines, why then not use only that approach? You could save yourself the effort of Case I. Please analyse and explain.

*Reply: This part of the paper was not written as clear as it can be. The terrain plane in this paper is an approximate terrain plane obtained by plane fitting using all triangulated 3D points. When one or two points near the line are available, the approximate terrain plane is shifted up using the one or two points, in order to obtain a better plane to approximate the real plane. In this paper, when Case II happens, this approximate terrain plane instead of a real fitted plane is used to estimate homography.*

*Compared with the method for Case I, the method for Case II is not as accurate and brings some wrong matches. In Table 4, it shows that the method for Case II Only will result in more wrong matches although it can generate more matches than the method for Case I. However, when only two or fewer 2D points are found, it is impossible to conduct matching by the method of Case I. Thus, the performance of Algorithm 1, which considers Case I and Case II at the same time, is better than those of Case II only and Case I only.*

*To prevent misunderstanding of readers, we have rephrased “the terrain plane” in the paper as “the approximate terrain plane”. We have now rewritten Section 4 to make the idea and process clearer. (See Page 12~14)*

Presentation:

-----

Some unsupported claims are made in the manuscript. Please either support your statements by conceptual arguments, or remove them.

(p4137) locally around a line an affine camera is a perfectly reasonable approximation of a perspective one. There is nothing that speaks against affine-invariant methods, unless you line is extremely long. Empirically you do get more lines than AILM, but there is no analysis or evidence that this is the extra lines are due to the use of homographies instead of affinities. Either show that it is so, or remove the claim that homographs are better.

*Reply: Now we have removed the claim.*

(p4119 and p5115) Again a cheap shot without a credible argument. First you say the

projective invariant method has high computational cost. Then you go on to explain that you exploit 3D points triangulated with bundle adjustment. Now bundle adjustment is really computationally expensive, and you have to include the time for computing those 3D points in the computational cost of your method (also in timing tables), since others who work without points can save that cost.

*Reply: We have now added the time for BA into the computational cost in Table 4. Because the BA algorithm used in the paper, ParallaxBA, can provide structure and motion estimation with small number of iterations and good efficiency, the computational cost for BA is not very high. (See Page 27, Table 4)*

Minor wording issues:

(p6l13) cameras do not use "film" anymore, please correct

*Reply: Done.*

(p13l36) it is not good practice to redefine math operators. The symbol  $\equiv$  denotes equivalence, not equality up to scale. Please follow the normal convention and use  $\propto$ .

*Reply: Done.*

# Line Matching based on Planar Homography for Stereo Aerial Images

Yanbiao Sun<sup>a,b</sup>, Liang Zhao<sup>b</sup>, Shoudong Huang<sup>b</sup>, Lei Yan<sup>a,\*</sup>, Gamini Dissanayake<sup>b</sup>

<sup>a</sup>*Institute of Remote Sensing and GIS, School of Earth and Space Science, Peking University, Beijing, 100871, China*

<sup>b</sup>*Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW 2007, Australia*

---

## Abstract

We propose an efficient line matching algorithm for a pair of calibrated aerial photogrammetric images, which makes use of sparse 3D points triangulated from 2D point feature correspondences to guide line matching based on planar homography. Two different strategies are applied in the proposed line matching algorithm for two different cases. When three or more points can be found coplanar with the line segment to be matched, the points are used to fit a plane and obtain an accurate planar homography. When two or fewer points can be found, the approximate terrain plane parallel to the line segment is utilized to compute an approximate planar homography. Six pairs of rural or urban aerial images are used to demonstrate the efficiency and validity of the proposed algorithm. Compared with line matching based on 2D point feature correspondences, the proposed method can increase the number of correctly matched line segments. In addition, compared with most of line matching methods without using 2D point feature correspondences, the proposed method has better efficiency although it obtains fewer matches. The C/C++ source code for the proposed algorithm is available on <http://services.eng.uts.edu.au/~sdhuang/research.htm>.

**Keywords:** Line Matching, Planar Homography, Point Feature

---

\*Corresponding author

Email addresses: syb51@pku.edu.cn (Yanbiao Sun), Liang.Zhao-1@uts.edu.au (Liang Zhao), Shoudong.Huang@uts.edu.au (Shoudong Huang), lyan@pku.edu.cn (Lei Yan), Gamini.Dissanayake@uts.edu.au (Gamini Dissanayake)

*Preprint submitted to Elsevier*

*November 17, 2014*

## 1. Introduction

Line extraction and matching is crucial for scene reconstruction, motion segmentation, structure from motion, robot navigation and so forth (Tang et al., 2006) (Schindler, 2006). For urban aerial images which consist of many man-made objects, line correspondences can help to reconstruct profiles of man-made objects, which are important elements for city reconstruction.

Extraction and matching of features, which mainly includes point features and line features, provides important data for many applications in photogrammetry, computer vision, etc. Compared with line feature extraction and matching, point feature extraction and matching is more mature with various approaches proposed (Ke and Sukthankar, 2004)(Novák et al., 2011)(Mikolajczyk and Schmid, 2005). Among them, Lowe proposes Scale Invariant Feature Transformation (SIFT), a robust and reliable approach invariant to transformation, scale, and illumination changes (Lowe, 2004). This method has been widely applied in many fields. To rapidly extract and match point correspondence from a pair of large-size aerial images,  $L^2$ -SIFT has been proposed (Sun et al., 2014), where a large aerial image is split into blocks and matching is conducted for every corresponding block using Sift-GPU (Wu, 2007). In this paper,  $L^2$ -SIFT will be used to extract and match 2D point correspondences, which are considered to be additional information for line matching.

Extracted line segments are unreliable for matching, because strong disambiguating geometric constraints are unavailable. Hence, line segment matching is significantly more difficult than point feature matching. During the last two decades, many approaches have been proposed to conduct line matching from stereo images. In 1997, Schmid and Zisserman proposed automatic line matching across views, which combined grey-level information and the multiple view geometric relations to achieve matching (Schmid and Zisserman, 1997). Each segment was treated as a list of points and point correspondences around the line segment were found using epipolar geometry. The average cross-correlation scores of all corresponding points acted as a similarity measure of the lines. In 2001, Heuel *et al.* used homogeneous vector and matrix to represent geometry entities and their uncertainties and then applied statistical tests to match and reconstruct 3D lines

without any given threshold (Heuel and Forstner, 2001). But this method can only handle with more than three images not stereo images. In 2005, to reconstruct DSM with a good quality, Zhang used cross-correlation method to match edges from linear array images (Zhang, 2005) with the help of the local geometric and photometric attributes. But, this method cannot be directly applied to process aerial images captured by a frame camera. In 2005, Bay *et al.* proposed automatic line matching for colour images, which obtained initial line correspondences, then further matches iteratively based on a topological filter (Bay et al., 2005). However, this method depended strongly on colour information and often fails processing remote sensing images. In 2009, Wang *et al.* proposed the mean standard deviation line descriptor method (MSLD) to achieve line matching. A line segment was regarded as a list of points and a histogram of the image gradient in a pixel support region was computed (Wang et al., 2009), and the mean and standard deviation of the histogram were obtained as line segment descriptors, similar to SIFT descriptors. In 2012, to improve matches from low-texture scenes, a method with appearance similarities and geometric constraints was proposed (Zhang and Koch, 2012). Direction histograms were used to estimate the perspective transformation and rotation angle and thereby reduce tentative matches. In 2012, Ok *et al.* used probability density functions and seven relational constraints to get more robust matches from urban aerial images (Ok et al., 2012). To improve efficiency and match accuracy, additional information, including DSM, LiDAR, GIS layer, surface details, etc., were used (Chehata et al., 2002) (Habib et al., 2013). In 2014, Gaussian mixture model and expectation maximization are used for line segment matching to achieve line segment registration using satellite images and GIS layer in (Long, 2014). The simple affine transformation used in Long’s approach may be good enough for satellite images, but may not be feasible for aerial images. In summary, the above methods heavily depend on grey-level and gradient information, and so will often fail to process images without strong disambiguating neighbouring line segments. They also have high computational complexity.

To improve efficiency and provide enhanced matches from low-texture images, some methods have been proposed using other information, including geometric properties, point correspondence, etc. In 1990, Stelmazyk attempted to use a nearest line strategy to guide line matching. Unfortunately, that was only suitable for image tracking, where images or line segments have relatively small changes (Stelmazyk, 1990). In 2009, Wang *et al.* proposed

the Line Signatures method to match from a wide-baseline image, where angle and length ratio between line segments are computed to describe a pair of line segments (Wang et al., 2009). However, this requires accurate location of the endpoints. In 2011, Elaksher proposed automatic line matching based on geometric properties, including the length of the perpendicular from the origin to the line, and the angle from the positive x-axis to the perpendicular (Elaksher, 2011). However, this method also requires grey-level properties and, therefore, is slow.

Considering that accurate point correspondences are easily obtained by some mature point feature extraction and matching algorithms, some researchers have proposed line matching methods built on point information. There are two representative papers for this approach. The projective invariant method for line matching using two lines and two points was proposed by Lourakis *et al.* (Lourakis, 2000). Unfortunately, this approach has a high computational cost. Two point-line invariant methods to process close-range images were proposed by Fan *et al.* (Fan et al., 2012). Similar to some methods based on geometrical information, the methods assumed the surrounding points were coplanar with the line in 3D space. One method based matching on the projective-invariant, which can be computed using four 2D image points and a line. The projective-invariant is regarded as a similarity measure between two tentative line segments and determines whether two line segments should be matched or not. In practical experiments, for some line segments, it may be difficult to find four points, and thus the total number of line matches will be reduced. The alternative proposed method relaxed the requirement of four points to only two points being required to compute the affine-invariant for a similarity measure. To increase line matches, we propose an approach based on planar homography. The software package based on the affine-invariant method (Fan et al., 2012) is publicly available and we compared those data to our proposed method.

Similar to the affine-invariant and projective-invariant methods, we assume that surrounding points are coplanar with the line segment in 3D space. Without assuming an affine camera, two different strategies are applied in the proposed line matching algorithm for two different cases. When three or more points coplanar with the line segment can be found, we fit a plane to the 3D points and calculate the projective transformation between two lines. When two or less points can be found, we use a parallel plane to calculate an approximate transformation. Because many line segments in aerial images are approximately parallel to the terrain plane, we consider the terrain

plane as the parallel plane. The 3D points, which are used to fit the plane containing the 3D line or the terrain plane, can be obtained from 2D point correspondences. Rather than using point-line invariant in 2D image space, 2D point correspondences are first triangulated into 3D points by Bundle Adjustment (BA).

Let us first define our notation. We express the camera projection matrix as  $P = K[R|t]$  or  $P = [A|a]$ , where  $K$  is the camera calibration matrix;  $R$  and  $t$  are camera rotation and translation, respectively;  $A = KR$ ; and  $a = Kt$ . A pair of images are considered to be source and target images,  $I_S$  and  $I_T$ , respectively. We use point feature extraction and a matching algorithm to obtain point feature correspondences,  $P_{2D}^F = \{x_{PS}^i, x_{PT}^i\}, i = 1, 2, \dots, N_P$ , where  $x_{PS}^i$  and  $x_{PT}^i$  are a pair of correspondences from the source image,  $I_S$ , and the target image,  $I_T$ , respectively; and  $N_P$  is the number of matched 2D features. The corresponding triangulated 3D points are  $P_{3D}^F = \{X^i\}, i = 1, 2, \dots, N_P$  in the local coordinate system. The set of line segments extracted from the source image is  $L_S = \{l_S^i\}, i = 1, 2, \dots, N_{LS}$ , where  $N_{LS}$  is the number of line segments. Each line segment is  $l_S^i = \{x_S^1, x_S^2\}$ , which includes two end-points,  $x_S^1$  and  $x_S^2$ . Similarly,  $L_T = \{l_T^i\}, i = 1, 2, \dots, N_{LT}$  expresses line segments from the target image. Lines in 3D space are  $L$ , where a matched line segment  $L^k = \{l_S^i, l_T^j\}, i \in \{1, 2, \dots, N_{LS}\}, j \in \{1, 2, \dots, N_{LT}\}$  includes two line segments of the two images. A 3D plane is  $P^L = [n_L^T, d]^T$ , where  $n_L$ , a  $1 \times 3$  vector, is the normal vector of the plane; and  $d$  is the depth information of the plane. Thus, the main focus of this paper is to find the corresponding line segment  $l_T^j \in L_T$  on the target image,  $I_T$ , when matching a putative line segment  $l_S^i \in L_S$  from the source image,  $I_S$ .

Section 2 briefly introduces planar homography, which provides an efficient tool to create the mapping of two corresponding line segments. Section 3 discusses the implementation of line matching based on an accurate projective transformation by fitting a plane using three or more triangulated 3D points when sufficient points are found. Section 4 discusses line matching based on an approximate transformation using the approximate terrain plane to match line segments when few points are found. Section 5 provides the complete algorithm, taking into account both cases, to obtain more matched line segments. In Section 6, six pairs of aerial rural and urban images are used to evaluate the proposed algorithm and compare with previous approaches. In Section 7 we draw our conclusions and discuss possible further studies.

## 2. Planar homography

The image points are formed by projecting the 3D target onto a camera, thereby discarding one dimension of information. The mapping between an arbitrary 3D point and its corresponding 2D image point can be represented with the  $3 \times 4$  camera projection matrix and two or more observations from different cameras are needed to estimate the depth information. In contrast to the grey-scale information, as in SIFT, mapping between image point correspondences on different cameras can be constructed only if their depth information is provided. However, if these 3D points are located on a plane, there is no loss of dimensionality, and the mapping can be achieved by a  $3 \times 3$  homogenous matrix, which is planar homography or projective transformation. Homography plays a significant role in calibration, metric rectification, scene reconstruction, tracking, etc. If the 3D plane is known, planar homography can be deduced by two camera projection matrices, as shown in Figure 1, where the plane is  $P^L = [n_L^T, d]^T$ , and  $x_1$  and  $x_2$  are the two image points corresponding to the same 3D point  $X$ . Suppose that the projection matrices are  $P_1 = K[I|0]$ , and  $P_2 = K[R|t]$ , for the two cameras, respectively, and the world coordinate frame is aligned with the first camera. The homography between image point correspondences is (Hartley and Zisserman, 2003)

$$H^d = K(R - \frac{tn_L^T}{d})K^{-1}. \quad (1)$$

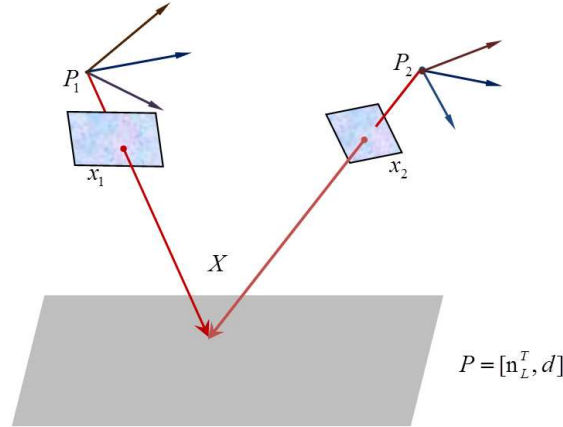


Figure 1: Planar homography can be computed accurately knowing the two camera projection matrices and a reference plane.

For some cases (e.g. the publicly available Aerial\_Large dataset (Werner, 2007)), only two projection matrices are provided. To compute homography in these cases, one way is to first decompose the projection matrix into calibration matrix, camera rotational matrix and translation, another way is to use the projection matrices directly. The formula of computing homography using two camera projection matrices is deduced below.

Let the first camera projection matrix be  $P_1 = [A_1|a_1]$ , and the second be  $P_2 = [A_2|a_2]$ . For simplicity, we may assume that the two images are captured by the same camera, thus the calibration matrices are the same. Suppose there is a virtual camera,  $P_0 = K[I|0]$ , and we may compute the homography  $H_1$  between  $P_1$  and  $P_0$  and  $H_2$  between  $P_2$  and  $P_0$  in Eq.(2). Then, the final homography between the first camera and the second camera can be computed by Eq.(3).

$$\begin{cases} H_1^d = (A_1 - \frac{a_1 n_L^T}{d})K^{-1} \\ H_2^d = (A_2 - \frac{a_2 n_L^T}{d})K^{-1} \end{cases} \quad (2)$$

$$H^d = H_2^d(H_1^d)^{-1} = (A_2 - \frac{a_2 n_L^T}{d})(A_1 - \frac{a_1 n_L^T}{d})^{-1}. \quad (3)$$

From Eq.(3), it is clear that the homography can be computed with only two camera projection matrices and a plane, without the need for the calibration matrix. We will use Eq.(3) to find the planar homography for two different cases.

### 3. Case I: Line matching based on a fitted plane

We discuss how to match a line segment when three or more surrounding 3D points can be found coplanar with the line segment. Our approach is based on a fitted plane using the corresponding 3D points. Thus, before line matching, 2D point feature correspondences are extracted by  $L^2$ -SIFT<sup>1</sup> and triangulated into 3D points using ParallaxBA<sup>2</sup> (Zhao et al., 2011)

---

<sup>1</sup> The  $L^2$ -SIFT source code can be downloaded at <http://services.eng.uts.edu.au/~sdhuang/research.htm>

<sup>2</sup> The ParallaxBA source code is available at OpenSLAM <http://openslam.org/ParallaxBA.html>

(Zhao et al., 2014). Two accurate camera projection matrices are also part of the output from ParallaxBA. Then, we can fit the plane containing the line segment using the 3D points. We treat every line segment as two end-points. Knowing two camera projection matrices and the fitted plane, we can create the mapping of the corresponding end-points of line segments from the source image to the target image. Rather than a direct comparison of the similarity between two line segments on original images, as per the projective-invariant or affine-invariant methods (Fan et al., 2012), candidate line segments on the target image are compared with a predicted line segment, formed by mapping the source line segment to the target image with planar homography. The process is detailed in the following subsections.

### 3.1. Finding coplanar 3D points around line segments

Without the depth information of the end-points, a line segment cannot be transformed into another image. However, if a 3D plane containing the line segment is known, planar homography can provide the mapping without the depth information. Thus, the plane plays an important role in mapping between lines appearing on different images and is essential to identify corresponding line segments. In the absence of other data, e.g. Digital Surface Model (DSM) or LiDAR point clouds, the 3D plane cannot be known, and we can only attempt to obtain 3D information from images. Fortunately, mature 2D point extraction and matching algorithms can generate 2D point correspondences, and we can use BA to transform these 2D points into 3D points. Although the number of 3D points are much fewer than for the point clouds from LiDAR or DSM, it only requires three 3D points to determine a plane and more 3D points can be used to fit a plane.

Consider the 3D scenario of the building in Figure 2. A 3D line segment,  $L$ , marked in red, is the edge of a building roof and is formed from the intersection of two side planes,  $P_L^l$  and  $P_L^r$ . Suppose that  $l_S$  and  $l_T$  are 2D projected line segments on the source image and the target image, respectively. Some triangulated 3D points are distributed on the two planes on each side of  $L$ , green points lie on  $P_L^r$  and red points are on  $P_L^l$ . For a given line segment,  $l_S$ , on the source image, we need to find the surrounding 2D point correspondences and separate them into two sets on each side of the 2D image coordinate.

Suppose a 2D line segment is  $l_S = \{x_{LS}^1, x_{LS}^2\}$ , where  $x_{LS}^1$  and  $x_{LS}^2$  are the two end-points. First, the center and the length of the line segment are calculated as  $M_{LS} = (x_{LS}^1 + x_{LS}^2)/2$  and  $d_{LS} = |x_{LS}^1 - x_{LS}^2|$ , respectively. Then

we traverse all potential points and choose a point if the distance to  $M_{LS}$  is less than  $d_{LS}/2$ . Since  $L^2$ -SIFT can generate many 2D point correspondences, to reduce searching time, K-D tree data structure is applied to create their indexes (Ooi et al., 1987). Once all eligible 2D point correspondences are chosen, their corresponding 3D triangulated points, which are regarded as coplanar points with 3D line, are found. Note that this assumption about point-line coplanarity will fail in some cases such as eaves of building and results in incorrect matches, because the points on one side are not coplanar with this line segment. In addition, because we use the 3D points on each side of the line to fit one plane, we divide the 3D points into two sets,  $C_{LS}^L$  and  $C_{LS}^R$ , located on the two sides. We achieve this using the 2D topology between image points and the line segment from the source images. If a 2D point feature lies on left of the line segment, the corresponding 3D point belongs to  $C_{LS}^L$ , and vice versa. If a point  $X^i$  lies on the line segment exactly, it will be assigned into both,  $X^i \in C_{LS}^L \cap C_{LS}^R$ .

Once  $C_{LS}^L$  and  $C_{LS}^R$  are found, we can fit the planes,  $P^L = [n_L^T, d]^T$ , by minimizing the distance between the 3D points and the plane,

$$(n_L, d) = \arg \min \sum \left( \frac{n_L^T X + d}{\sqrt{n_L^T n_L}} \right), X \in C_{LS}^L \cup C_{LS}^R. \quad (4)$$

We used the RANSAC package to fit the planes (Engels et al., 2008). In Figure 2, the blue plane  $P_L^l$  and the green plane  $P_L^r$  are fitted using  $C_{LS}^L$  and  $C_{LS}^R$  when three or more 3D points are found.

### 3.2. Calculating the homography and finding candidate line segments

Once the camera projection matrices and 3D plane are known, we calculate the planar homography  $H^d$  from Eq.(3). A given line segment on the source image is mapped to a predicted line segment on the target image. In Figure 3, a predicted line segment  $l_E = \{x_{LE}^1, x_{LE}^2\}$ , corresponding to the line segment  $l_S = \{x_{LS}^1, x_{LS}^2\}$  in red, can be obtained by simple projective transformation, and the two end-points of the line segment are transformed,

$$\begin{cases} x_{LE}^1 = H^d x_{LS}^1 \\ x_{LE}^2 = H^d x_{LS}^2 \end{cases} \quad (5)$$

In Figure 3, blue points are chosen to fit the plane and calculate the homography  $H^d$ . We obtain a predicted line segment, shown in blue on the target image, and the black point is the center of that predicted line segment.

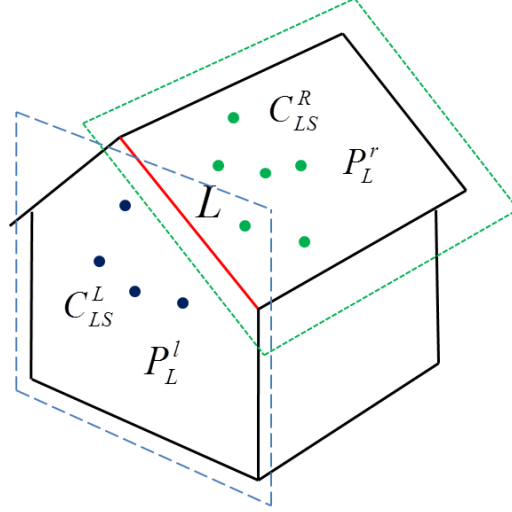


Figure 2: Two fitted planes, derived using RANSAC, where the blue and green points were distributed on either side of the red line.

Now we need to find all candidate line segments on the target image close to the predicted line segment. In this paper, to simplify the algorithm, a line segment will be chosen as a candidate one using a threshold instead of using statistical tests by computing the uncertainty of the line. When the distance between its center and center of the predicted line segment is less than a given threshold, a line segment will be regarded as a tentative line segment. We chose the threshold  $1.5 \times$  the length of the predicted line segment. Thus, (see Figure 3) the three line segments in yellow are discarded, and  $T_L = \{l_T^1, l_T^2, l_T^3\}$  is the set of candidate line segments.

We also consider the ordering constraint of the 2D topology between points and lines to reduce the number of false positives and mismatches, although it will fail to process thin lines such as street lamps. The blue 2D point correspondences locate on the left of the chosen line segment on the source image, so  $l_T^2$  should be removed from  $T_L$ . The new candidate line segments are  $T_L^{new} = \{l_T^1, l_T^3\}$ , shown in black in Figure 3. Now we need only measure the similarity between our projected line segment,  $l_S$ , and the candidate line segments,  $T_L$ , to match the line segment corresponding to  $l_S$ .

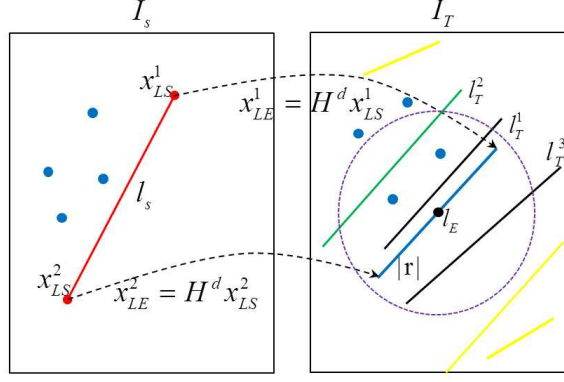


Figure 3: Predict a line segment and find all possible candidate line segments. Blue points are chosen to fit a plane and obtain the planar homography. Given a source line segment on  $I_S$  (red), we obtain a predicted line segment (blue) and find candidate line segments (green and black ) on  $I_T$ .

### 3.3. Finding the matched line segment from the set of candidate line segments

For a candidate line segment,  $l_T = \{x_{LT}^1, x_{LT}^2\}$ , on the target image, if  $l_T$  and  $l_E$  are the most similar pair, then  $l_T$  will be the final matched line segment of  $l_S$ . To find the matched line segment from the set of candidates, the similarity between each candidate and the projected line segment is assessed by two aspects: orientation and position. The orientation is the angle between the predicted line segment  $l_E$  and the candidate,  $l_T^i$ , shown as  $\theta$  in Figure 4,

$$\theta = \arccos \frac{l_T^{iT} l_E}{|l_T^i| |l_E|} . \quad (6)$$

The position shift is the mean of the distances from the end-points of the candidate to the predicted line segment,

$$\rho = \frac{1}{2}(d_1 + d_2) = \frac{1}{2} \left( \frac{|x_{LT}^{1T} l_E|}{l_E^T l_E'} + \frac{|x_{LT}^{2T} l_E|}{l_E^T l_E'} \right) \quad (7)$$

where  $l_E'$  is the first two parameters of a 2D line.

To make the algorithm simpler, if the orientation and position shift are less than given thresholds, the candidate will be regarded as the line segment corresponding to the original source line segment. When there are multiple possible matches whose similarities are greater than a given threshold, the

method can generate one-to-many matches if only the threshold is applied. To reduce the number of wrong matches, the tentative line with maximum similarity will be regarded as the final match in this paper. If three or more 2D point correspondences can be found simultaneously on each side of the line segment, we must fit two planes, hence deriving two matched line segments. We then choose the candidate line segment with the least position shift to be the match.

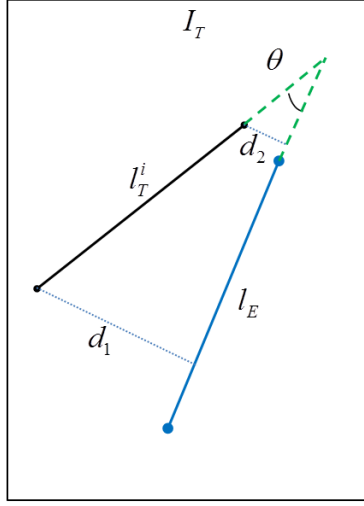


Figure 4: Similarity between a candidate line segment and a predicted line segment: orientation and position shift.

#### 4. Case II: Line matching based on the approximate terrain plane

When only two or fewer 2D points on one side can be found coplanar with the line segment, we propose to use the approximate terrain plane to compute homography. This method can obtain matches for some line segments which cannot be matched using the method in Section 3.

For aerial photogrammetric cases, objects on the ground and roofs of buildings are the main structure elements. Thus, many lines of the objects on the ground and edges of the roofs are either located on the terrain plane or are parallel to that plane. Since we have no idea of the accurate terrain plane in advance, in order to find a plane which edges of building are parallel with in aerial study case, we fit a plane parallel to the terrain plane using all triangulated 3D points. This plane is very close to the terrain plane and is

called the approximate terrain plane,  $P_G = [n_G^T, d]$ , without loss of generality. Note that the method proposed in this section will not be able to address the red line in Figure 2, because it is not parallel to the plane. Once the terrain plane has been determined, line matching follows the same process as in Section 3. Although the error of the predicted line segment using the approximate plane will not be large in aerial study case, it is still larger than that with the fitted plane in Case I. Hence, the threshold of the position shift to determine whether a candidate line segment is accepted should be slightly larger than that in Case I.

To analyse the accuracy of the predicted line segment on the target image when the approximate plane is used, let us assume that the camera projection matrix of the source image is  $P_1 = K[I|0]$ , the projection matrix of the target image is  $P_2 = K[R|t]$ , where the world coordinate frame is defined by the camera of the source image. A 3D line is parallel, or lies in, a plane  $P_L = [n_L^T, d]^T$ , and is projected into a 2D line segment  $l_S = \{x_l^1, x_l^2\}$  on the source image. A predicted line segment  $l_E = \{x_E^1, x_E^2\}$  can be obtained from

$$l_E \propto (K^{-T}RK^T + \frac{K^{-T}Rn_Lt^TRK^T}{\pm h\sqrt{n_L^Tn_L - n_L^TR^Tt}})(x_l^2 \times x_l^1) \quad (8)$$

where  $\propto$  indicates equality up to scale, and  $h = d/\sqrt{n_L^Tn_L}$ . The detailed derivation of Eq. (8) is given in Appendix. Then, we may express the predicted line segment using the depth information of the plane relative to the first camera. Consider Figure 5,  $O$  is the origin of the coordinate frame, and  $P_G^L$  is a virtual plane containing the 3D line segment  $L$ , which is parallel to a plane  $P_G$ . The distance between  $O$  and  $P_G$  is  $h_G$ , while the distance between  $O$  and  $P_G^L$  is  $h_L$ , and  $h_B$  indicates the distance of two parallel planes. If we use the parallel plane  $P_G$  rather than the plane  $P_G^L$  to compute the predicted line segment, Eq.(8) can be written as

$$l_E \propto (K^{-T}RK^T + \frac{K^{-T}Rn_Lt^TRK^T}{\pm(1+\delta)h_L\sqrt{n_L^Tn_L - n_L^TR^Tt}})(x_l^2 \times x_l^1) \quad (9)$$

where  $h_B = \delta h_L$ , for some constant,  $\delta$ .

Now we use the plane  $P_G$  instead of the plane  $P_G^L$  to get a predicted line. For a given camera and known normal of the plane, the translation between two cameras can be set as one (to scale). When matching a line segment which lies on the plane  $P_G^L$  using Eq.(9), the error of the predicted

line segment depends on the rotational matrix and the constant  $\delta$ . If  $\delta = 0$ , the error of the predicted line segment is zero. For an aerial photogrammetric scene, the height of a building is in general very low compared to the height of the camera platform, and so  $\delta$  is small. For the urban scene,  $\delta \leq 0.05$ . For the rural scene,  $\delta$  is even smaller. Thus, the error of the predicted line segment is expected to be small. In Section 6.2, we will use simulation and real data to quantify the error of the predicted line segment arising from the rotation matrix and  $\delta$ .

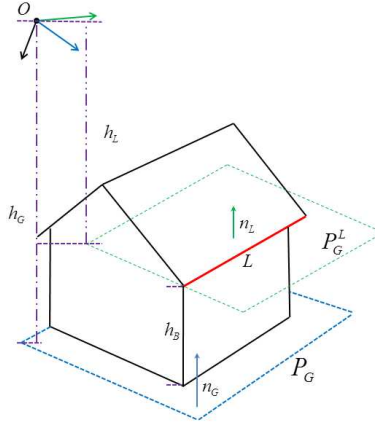


Figure 5: There is not enough surrounding points to fit a plane. We use the approximate terrain plane,  $P_G$ , to substitute for the fitted plane,  $P_G^L$ , to realize line matching.

In addition, from Eq.(9), if we impose the approximate terrain plane close to the real plane or a virtual plane, i.e.,  $\delta \approx 0$ , we can improve the accuracy of the predicted line segment. For some line segments where one or two points are found, the points cannot be used to fit a plane, but can be utilized to improve the accuracy of the predicted line segment by adjusting the depth information of the plane. We can calculate a center point of these 3D points, and then compute the new  $d$  in Eq.(1) by forcing the center point to be located on the parallel plane. This will increase the accuracy of line matches for Case II and will be applied for the general algorithm in Section 5.

## 5. The general algorithm, combining the two cases

We provide an algorithm (Algorithm 1) to incorporate the two cases of Section 3 and Section 4, which can match more line segment correspondences with a high accuracy.

Before line matching,  $L^2$ -SIFT is utilized to extract and match 2D point correspondences,  $P_{2D}$ , from a pair of aerial images. These 2D point correspondences are input to ParallaxBA, and 3D points,  $P_{3D}$ , are triangulated, as well as calculation of the camera projection matrices,  $P_S$  and  $P_T$ . To decrease search complexity, two K-D tree data structures are constructed, where their nodes are 2D point correspondences on the source image and line segments extracted on the target image, respectively.

We then match all the line segments that meet the condition of Case I. First, we initialize a set of line segments  $\tau$ , which stores failed line segments by the method in Section 3. Second, for each line segment,  $l_S^i$ , on the source image among  $L_S$ , we calculate the center and length of  $l_S^i$ , to find all surrounding 2D point correspondences on both sides which are coplanar with the line segment. If the numbers of points on both sides ( $m_1, m_2$ ) are less than three, matching will be stopped and the line segment  $l_S^i$  is stored into  $\tau$ . If not, 3D points on both sides are found and fitted as two planes, respectively. Third, for each plane, we calculate the planar homography, obtain a predicted line segment on the target image, find all candidate line segments around this, and calculate the orientation and position shift. A candidate line segment with the least position shift is chosen for each side. The line segment with least position shift, where the position shift is less than a given threshold,  $\rho_g^1 = 5$  pixels, will be regarded as the final corresponding line segment to be output to  $L^k$ . If the position shift exceeds  $\rho_g^1$ , the source line segment is also stored into  $\tau$ .

Finally, we match the rest of the line segments, which could not be matched above. First, we fit the approximate terrain plane  $P_G$  with all triangulated 3D points. Second, we traverse each line segments in the holding set,  $\tau$ . We find all 2D point correspondences around the line segment. If not found,  $P_G$  takes the place of  $P_G^L$  directly. If some points are found, a center point is calculated, and we adjust the depth information of the parallel plane by forcing the center point to be located on the plane. Third, we follow the same operations, including calculating the homography, obtaining the predicted line segment on the target image, finding all candidate line segments, and calculating orientation difference and position shift. The correct match also should meet the condition that its error is less than the threshold  $\rho_g^2 = 20$  pixels.

---

**Algorithm 1** Line matching based on planar homography for stereo aerial images

---

**Input:** two sets of line segments from the source image and the target image,  $L_S$  and  $L_T$

**Output:** a set of matched line segments  $L^k = \{l_S^i, l_T^j\}$

1: **Preliminary**

2: extract and match 2D point correspondences  $P_{2D}$  by  $L^2$ -SIFT

3: do bundle adjustment (ParallaxBA) to triangulate 3D points  $P_{3D}$  and calculate two camera projection matrices,  $P_S = [A_S|a_S]$  and  $P_T = [A_T|a_T]$

4: construct two K-D tree for 2D points on the source image and line segments  $L_T$  on the target image

5: **Case I: line matching based on a fitted plane**

6: initialization  $\tau = \emptyset$  (empty set)

7: **for** each  $l_S^i \in L_S$  **do**

8:   calculate the center and length of  $l_S^i$ ,  $M_{LS} = (x_{LS}^1 + x_{LS}^2)/2$  and  $d_{LS} = |x_{LS}^1 - x_{LS}^2|$

9:   find surrounding 2D points on each side of  $l_S^i$  in  $P_{2D}$ , the number of points are  $m_1$  and  $m_2$ , respectively.

10:   **if**  $m_1 < 3$  and  $m_2 < 3$  **then**

11:      $l_S^i \in \tau$ , continue

12:   **end if**

13:   find corresponding 3D points,  $X \in C_{LS}^L$  or  $C_{LS}^R$

14:   **for** each side consisting of at least 3 points **do**

15:     fit plane with RANSAC using  $C_{LS}^L$  or  $C_{LS}^R$

16:     calculate homography with the fitted plane

17:     obtain a predicted line segment  $l_E$

18:     find the candidate line segments  $T_L$

19:     remove line segments with large orientation differences

20:     calculate position shift

21:     find line segment with least position shift  $\rho_1$  and  $\rho_2$  on each side

22:   **end for**

23:   choose least position shift  $\rho'$  between  $\rho_1$  and  $\rho_2$

24:   **if**  $\rho' < \rho_g^1$  **then**

25:      $\{l_S^i, l_T^j\} \in L^k$

26:   **else**

27:      $l_S^i \in \tau$

28:   **end if**

29: **end for**

---

---

```

30: Case II: line matching based on the terrain plane
31: fit the approximate terrain plane  $P_G = [n_G^T, d]$  with all points  $P_{3D}$ 
32: for each  $l_S^i \in \tau$  do
33:   find surrounding 3D points  $\sqsupset$  around  $l_S^i$ 
34:   if  $\sqsupset = \emptyset$  then
35:      $P_G^L = P_G$ 
36:   else
37:     calculate the center of  $\sqsupset$ , as  $C_{\sqsupset}$ 
38:      $n_G^T C_{\sqsupset} + d' = 0$ 
39:      $P_G^L = [n_G^T, d']$ 
40:   end if
41:   do the same operation as line 16  $\sim$  21
42:   calculate the least position shift  $\rho'$ 
43:   if  $\rho' < \rho_g^2$  then
44:      $\{l_S^i, l_T^j\} \in L^k$ 
45:   end if
46: end for

```

---

## 6. Experimental results

Six pairs of rural and urban images are used to check the validity and accuracy of the proposed line matching algorithm. The accuracy of the predicted line segment is very critical for the proposed method, and affects the accuracy of matching. Thus, to demonstrate the feasibility of line matching using parallel plane for aerial scenes, we must first investigate the accuracy of the predicted line segment affected by camera pose and  $\delta$  in Eq.(9), using simulations and real experimental data. We then conduct line matching using six datasets following the strategies of Algorithm 1 in Section 5. The results are compared with those of the line matching method based on affine-invariant (Fan et al., 2012), OK's method (Ok et al., 2012), Schmid's method (Schmid and Zisserman, 1997) and MSLD method (Wang et al., 2009), with the same unmatched line segments from two images. We have focused on line matching, rather than line segment extraction, so the final line segment extraction was done using publicly available Line Segment Detector (LSD) software (Grompone von Gioi et al., 2012).

	Aerial_Large	DunHuan	Tongzhou	Guanzhou	Toronto	Vaihingen
Camera	Unknown	Canon	Canon	Sony	UCD	DMC
Scene	Rural	Rural	Rural	Urban	Urban	Urban
Image Size	$2000 \times 2000$	$5616 \times 3744$	$4272 \times 2848$	$4592 \times 3056$	$11500 \times 7500$	$7680 \times 13824$
Flight Height(m)	Unknown	350	500	200	1600	900
Average Height of Object (m)	Unknown	4	5	10	20	10
GSD (m)	Unknown	0.1	0.15	0.07	0.15	0.08
Overlap	95%	60%	60%	75%	60%	60%

Table 1: Overview of the test datasets.

### 6.1. Data tests

Detailed parameters of the six pairs of aerial images are shown in Table 1. All of the images were taken using high-quality cameras equipped on aircraft platforms and comprise many objects, whose edges are parallel to the terrain plane. Among these datasets, the publicly available Aerial\_Large dataset can be downloaded from LMatch (Werner, 2007); and the publicly available Toronto and Vaihingen dataset are from the ISPRS Test Project on Urban Classification and 3D Building Reconstruction (Cramer, 2010) (Rottensteiner et al., 2012) (Rottensteiner et al., 2013). To demonstrate the result clearly, for first five datasets, a pair of patch  $500 \times 500$ , is used in Section 6.3.1. In addition, the fifteen patches of Vaihingen provided by (Ok et al., 2012) are used in Section 6.3.2. It is demonstrated by experiments that switching the source image and the target image results in very similar matches, so we choose any one of them as the source image and another one as the target image.

### 6.2. The accuracy of the predicted line segment calculated from parallel planes

In our proposed algorithm, matching accuracy will depend heavily on the accuracy of the predicted line segment. For a line segment on the source image when three or more points can be found to be coplanar, a fitted plane containing the line segment can provide accurate planar homography, and so the accuracy of the predicted line segment is very high. For line segments lacking sufficient coplanar points, planar homography based on the approx-

Parameters	Value
Camera Size (pixels)	$5000 \times 4000$
Focal Length (pixels)	3000
Rotational Angles (degrees)	$[0, 0, 0]^T$
Translation	$[1, 0, 0]^T$

Table 2: Scenario parameters of simulation experiments

imate terrain plane is approximated and incorporates an orientation difference and position shift of the predicted line segment. This section analyses the accuracy of the predicted line segments using different  $\delta$  in Eq.(9) with simulation and real experimental data, and demonstrates that the method proposed in Section 4 is feasible for aerial images.

#### 6.2.1. Accuracy using simulation

We analyse the accuracy of the predicted line segment with respect to the rotation matrix and height ratio,  $\delta$ . We first fix camera poses and change  $\delta$ , then allow camera poses and  $\delta$  to vary simultaneously.

We assume the scenario shown in Figure 6, with the parameters as listed in Table 2, and that the internal parameters of the camera are very similar to those used in the DunHuan dataset (without distortion). Three rotational angles are zero and there is only X-direction translation, which is one (to scale). We define a local coordinate system where the origin is the first camera pose  $C_1$  in Figure 6. The three rotational angles, yaw, pitch and roll;  $\alpha, \beta, \gamma$ ; are rotations about the Z, Y, and X axis, respectively.

The red line segment in Figure 6 is to be matched. It lies on the green plane in Figure 6. The distance between the origin of the local coordinate and the green plane is  $H$ , while  $h$  is the distance between two different parallel planes. The main task is to match this line segment located on the plane with  $n = [0.15, 0.09, -1]^T$ ,  $H = 2.9$ . When we assume there is no noise, for the line segment on the left image, the predicted line segment  $l_E^0$  on the right image by the green plane is exact and is regarded as the benchmark. Changing  $h = \delta H$ , we can use a new parallel plane to obtain a new approximate predicted line segment  $l_E^h$ , where the new depth of the parallel plane is now  $(1 + \delta)H$ , and orientation difference and position shift can be calculated using  $l_E^h$  and  $l_E^0$ . We set  $\delta = 0, 0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3$ , and  $0.5$ . When  $\delta = 0$ , the predicted line segment,  $l_E^0$ , is exact without any position shift and orientation difference, which is used for assessing the accuracy by using different parallel

planes.

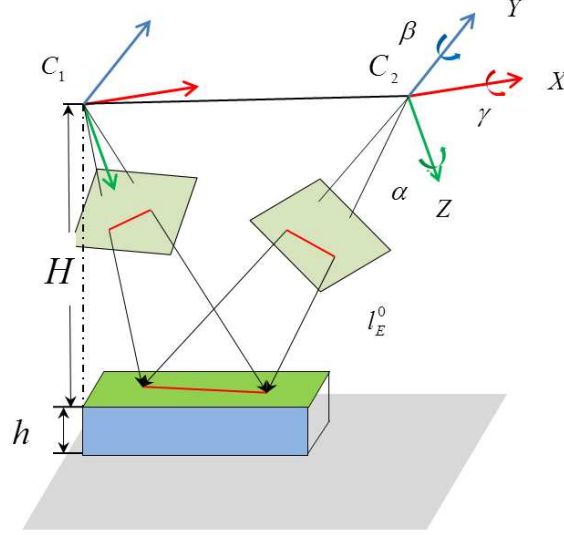


Figure 6: A 3D line segment in red on the green plane projects into two cameras, with baseline 1,  $H=2.9$ . The green plane is the real plane containing the 3D line, and the grey plane is the approximate terrain plane to map a line segment on the left image into a corresponding line segment approximately, where the distance between the origin of the coordinate system and the parallel plane is  $H + h$ . If  $h = 0$ , an exact predicted line segment on the right image is obtained.

When the rotational angles are zero, the orientation difference and position shift with respect to  $l_E^0$  for different  $\delta$  are shown in Figure 7, where Figure 7(a) is the orientation difference and Figure 7(b) is the position shift. The orientation difference is less than one degree, whereas the position shift increases with  $\delta$ . In this case, when  $\delta < 0.05$ , the position shift is within 25 pixels.

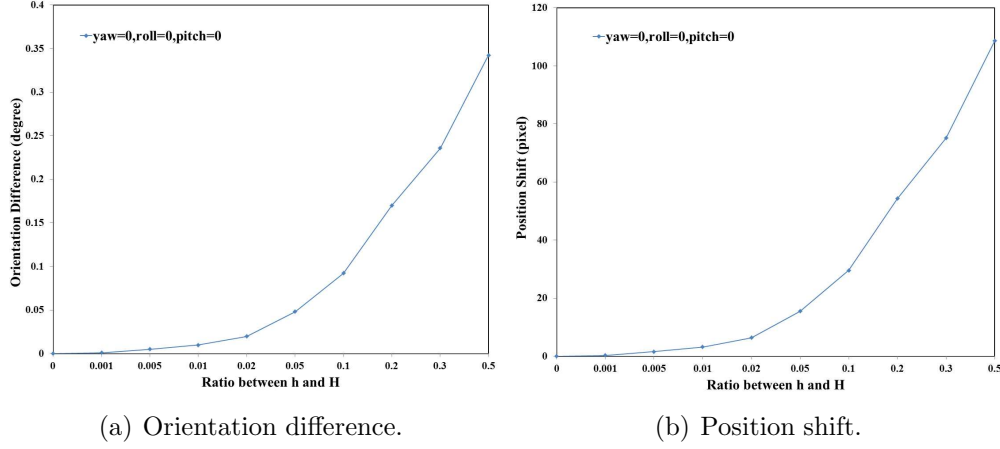


Figure 7: The orientation difference and position as a function of the ratio between  $h$  and  $H$  when the rational angles are zero.

In practical flight, the roll and pitch angles cannot remain zero, and the yaw angle particularly can rotate arbitrarily. We use the same parameters of Table 2, and analyse orientation difference and position shift respectively when one angle varies while the other two remain zero.

First, we fix the roll and pitch as zero, and allow the yaw to vary as 0, 45, 90 and 180 degrees (Figure 8). Clearly, orientation difference and position shift are independent of the yaw angle.

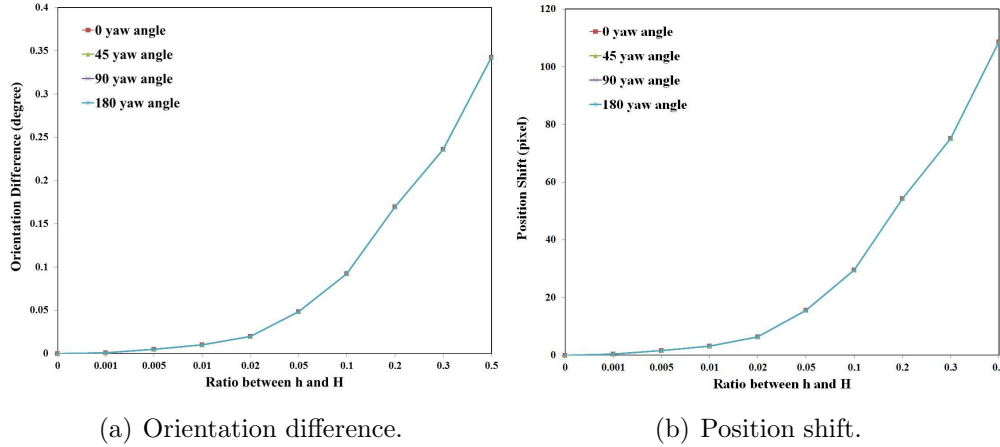


Figure 8: The orientation difference and position shift as a function of the ratio between  $h$  and  $H$  with fixed (0) roll and pitch and varying yaw.

Next, we fixed yaw and pitch to zero, and allowed roll to be 0, 5, 10, 15, 20, 25 and 30 degrees (From Figure 9). Although the orientation difference and position shift increase with the roll angle, the maximum orientation difference is less than one degree (Figure 9(a)) and position shift remains  $\leq 25$  pixels when  $\delta < 0.05$  (Figure 9(b)).

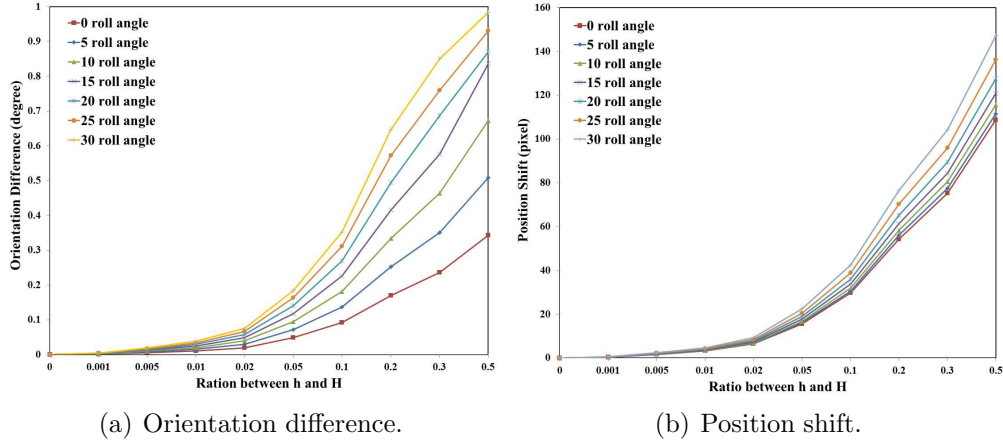


Figure 9: The orientation difference and position shift as a function of the ratio between  $h$  and  $H$  with fixed (0) pitch and yaw and varying roll.

Finally, we fixed yaw and roll at zero and allowed the pitch to vary up to 30 degrees (Figure 10). The orientation difference increases with pitch angle, but remains  $\leq 1$  degree and can be ignored (Figure 10(a)). In addition, position shifts  $\leq 25$  pixels when  $\delta < 0.05$  (Figure 10(b)).

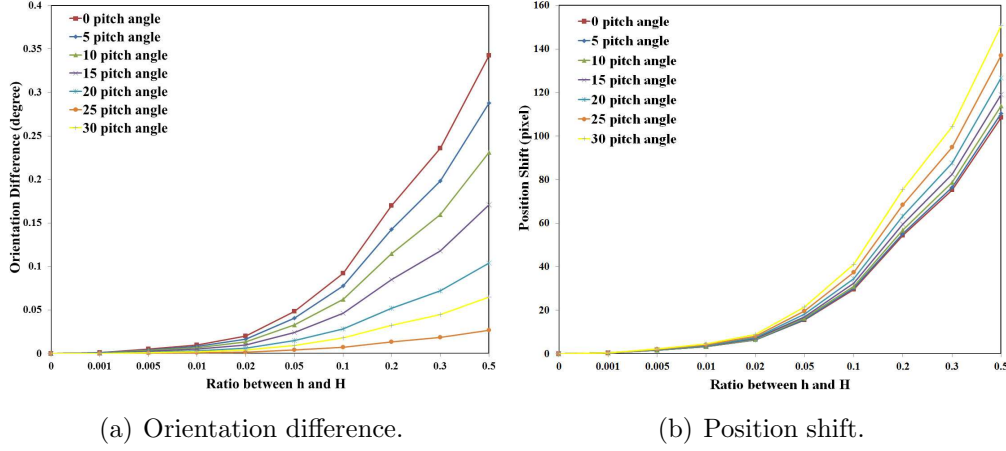


Figure 10: The orientation difference and position shift as a function of the ratio between  $h$  and  $H$  with fixed (0) yaw and roll and varying pitch.

It is clear that the rotation effect on orientation difference is small and can be ignored when we use the approximate terrain planes to get a predicted line segment. Although position shift increases with  $\delta$ , it remains always less than 25 pixels when  $\delta = 0.05$ . In most practical aerial photogrammetry, to survey economically and efficiently, the platform is high, whereas the height of an object on the terrain is relatively small and so in general  $\delta < 0.05$ . Thus, line matching using a parallel plane is feasible for most aerial images.

### 6.2.2. Accuracy using real data

We discuss the accuracy of the predicted line segment using real data when the yaw, roll and pitch angles are not zero. In this section, we use a given line segment from images in the Guanzhou dataset, where the rotational angles of the second camera with respect to the first camera are 0.77, 0.13 and -0.42 degrees.

First, we find a pair of corresponding line segments manually on two images and three red points locating on the plane. Figure 11 is a part of the target image of the Guanzhou dataset, where the predicted line segments using the ratio from 0 to 0.5 are drawn (the red line when  $\delta = 0$ ). The predicted lines are shown as different colours and numbers. The predicted line is very close to the red line when  $\delta < 0.05$ , but diverges as  $\delta$  increases. The orientation difference and position shift are shown in Figure 12. In contrast to the simulation, when the  $\delta = 0$  the orientation and position shift are not zero, because there are small errors in the estimated 3D points and

camera projection matrices. However, the orientation difference remains  $< 1$  degree regardless of  $\delta$ , and thus can be ignored. The height of the building roof in this scenario is about 10 meters and flight height is about 200 meters, hence  $\delta \approx 0.05$ , and the position shift is approximately 25 pixels.

There is no doubt that the method will fail for some line segments when  $\delta$  is large. When the height of imaged objects is large or the platform low, large position shift is inevitable. However, one or two points around some line segments, which are not enough to fit a plane, can be used to estimate the depth of plane and thus result in a smaller position shift to improve matching accuracy. This is implemented in the proposed algorithm of Section 5.

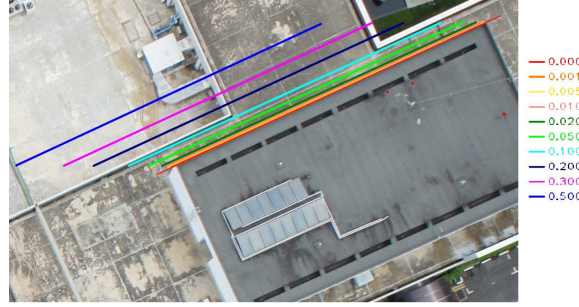


Figure 11: Three red points on the roof are chosen to calculate an accurate plane to obtain a correctly matched line segment in red,  $\delta = 0$ . Changing  $\delta$ , other predicted lines are obtained, which are drawn with different colours and labels. The predicted line segments with larger  $\delta$  are distributed further away from the red line segment.

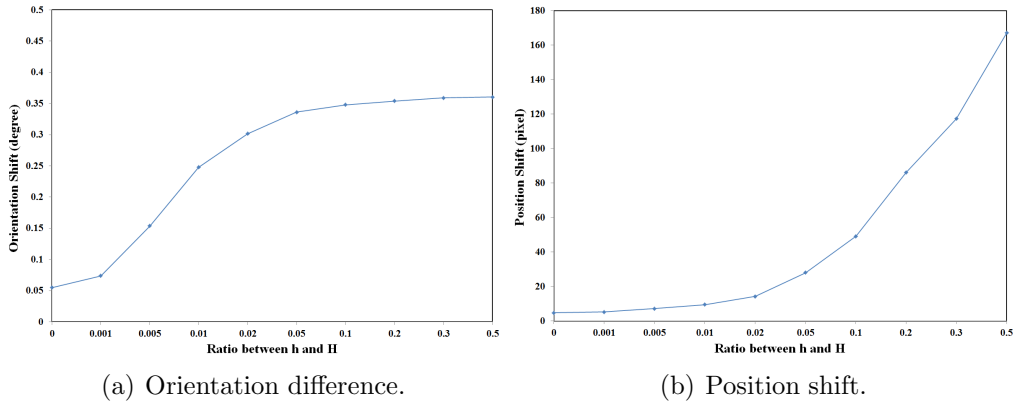


Figure 12: For a given line segment from the Guanzhou dataset, the orientation difference and position as a function of the ratio between  $h$  and  $H$ .

### 6.3. Line matching results

In this section, our proposed method will be compared with two kinds of methods. First, in Section 6.3.1, line matching method based on affine invariant of point-line (Fan et al., 2012), which also use 2D point feature correspondences, will be compared. Second, in Section 6.3.2, Schmid’s method (Schmid and Zisserman, 1997), OK’s method (Ok et al., 2012) and MLSD method (Wang et al., 2009), which conduct line matching without using 2D point feature correspondences, will also be compared.

#### 6.3.1. Comparison with a method using 2D point feature correspondences

In this section, we present results from Aerial\_Large, DunHuan, Tongzhou, Guanzhou and Toronto datasets using three methods:

- Case I Only: We match line segments using only the strategy of Case I. When three or more points are found to be coplanar with the line segment, we match the line segment using the accurate planar homography. When only two or fewer points can be found, the line segment is abandoned.
- Case II Only: We match line segments using only the strategy of Case II. No point is used when matching each line segment. A plane should be given in advance. We manually choose three points on the ground to form a plane.
- Algorithm 1 (Section 5): We first match line segments when three or more points are found to be coplanar with the line segment. For those line segments where two or fewer points can be found, it is matched using the terrain plane.

With the same input of unmatched line segments and 2D point correspondences, the results of the three methods are compared with those using the publicly available software package based on affine-invariant method (Fan et al., 2012), which we have labelled here as AILM.

In Table 3, we list a number of line segments extracted by LSD from the source and target images, whose lengths are larger than a given threshold. According to their GSD, we set the threshold as 15, 15, 20, 10 and 20 pixels for the Aerial\_Large, DunHuan, Tongzhou, Guanzhou and Toronto datasets, respectively. The number of 2D point feature correspondences extracted by  $L^2$ -SIFT is also given. To compare match completeness, we manually

	Aerial_Large	DunHuan	Tongzhou	Guanzhou	Toronto
# Line (Source)	89	129	175	260	172
# Line (Target)	86	125	167	170	165
# point	121	94	60	67	210
# Maximum Matches	62	101	129	142	135

Table 3: Number of extracted line segments and points and maximum matches.

counted the maximum matches. The number of total matches, the number of incorrect and correct matches, correctness and computational time are listed in Table 4, and completeness of matches is shown in Figure 13. Matches correctness was assessed manually, following the same strategy as (Schmid and Zisserman, 1997) (Fan et al., 2012) (Ok et al., 2012). All tests were executed on an Intel CPU i5-760 computer on the Windows platform with a 2.5 GHz CPU.

Correctness is similar and high for Case I Only and AILM. The number of matches is also very similar for those methods.

Case II Only in general matches more lines than either AILM or Case I Only, but has more incorrect matches, due to the inexact depth information of the plane causing position shift and orientation differences in the predicted line segment. However, the correctness can reach  $\geq 50\%$ . For rural images, including Aerial\_Large, DunHuan, and Tongzhou datasets, the line segments are very close to the terrain plane,  $\delta$  is small, and thus the correctness of Case II Only  $\approx 75\%$ . For urban datasets, the correctness is lower because some line segments are far from the terrain, hence  $\delta$  is large. Thus, it is inevitable to get incorrect matches when lacking information.

Algorithm 1 not only increases the number of matches compared with AILM and Case I Only, but also achieves higher correctness than Case II Only. This is because Algorithm 1 combines the advantages of Case I Only and Case II Only. Furthermore, our proposed method is as fast as AILM.

Algorithm 1 can efficiently achieve matching line segment with a good accuracy and higher completeness, and so would be suitable line matching for aerial photogrammetric images.

In Figures 14(a) ~ 18(a), we show the matched line segments by AILM, whereas Figures 14(b) ~ 18(b) show the matches by Algorithm 1. The corresponding matched line segments on two images are drawn with green colour. Red line segments represent incorrect matches.

Dataset	Test	# Total Matches	# Wrong Matches	Correctness	#Correct Matches	time (sec)
Aerial_Large	AILM	28	6	78.5%	22	0.15
	Case I Only	32	7	78.1%	25	0.2
	Case II Only	32	8	75%	24	0.2
	Algorithm 1	56	11	80.3%	45	0.25
<i>DunHuan</i>	AILM	30	1	96.6%	29	0.18
	Case I Only	28	0	100%	28	0.22
	Case II Only	65	7	89.2%	58	0.23
	Algorithm 1	70	6	91.4%	64	0.25
<i>Tongzhou</i>	AILM	19	1	94.7%	18	0.12
	Case I Only	22	0	100%	22	0.2
	Case II Only	92	6	93.4%	86	0.2
	Algorithm 1	82	3	96.3%	79	0.25
<i>Guanzhou</i>	AILM	27	4	85.2%	23	0.15
	Case I Only	50	6	88%	44	0.2
	Case II Only	88	37	57.9%	51	0.2
	Algorithm 1	84	13	84.5%	71	0.25
<i>Toronto</i>	AILM	76	6	92.1%	70	0.4
	Case I Only	72	5	93.1%	67	0.3
	Case II Only	125	54	56.8%	71	0.25
	Algorithm 1	123	13	89.4%	110	0.35

\*The time of proposed methods includes computational cost of BA and matching.

Table 4: Comparison between the proposed approach and the affine-invariant method (AILM).

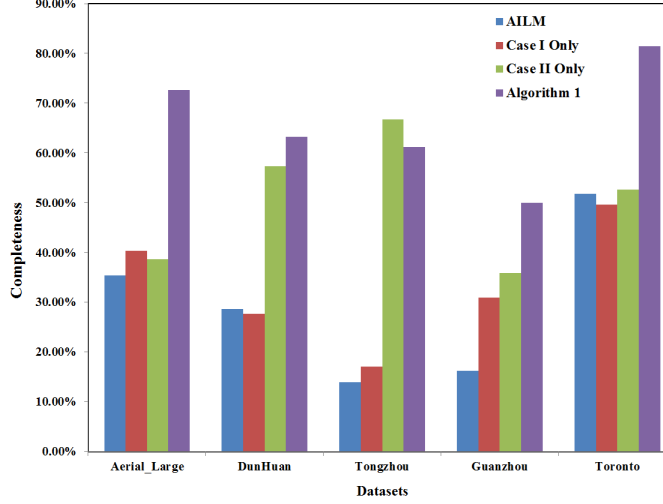


Figure 13: The completeness of correct matches using AILM, Case I Only, Case II Only and Algorithm 1.

### 6.3.2. Comparison with state-of-the-art methods without using 2D point feature correspondences

In this section, three state-of-the-art methods, which conduct matching using grey-level information and other geometric constraint rather than using 2D point feature correspondences, are compared with our proposed method.

First, with fifteen patches of Vaihingen dataset, we present matches completeness using OK's method, Schmid's method and our proposed method in Figure 19. Besides, the elapsed time for matching using four patches of them are listed in Table 5. The matches completeness and the elapsed time of OK's method and Schmid's method are copied from (Ok et al., 2012). In addition, the elapsed time of OK's method and Schmid's method is measured on a laptop computer with a CPU Intel Core2 Duo 2.53 GHz and 4GB RAM, which is very similar to the computer used to run our proposed method. From Figure 19, Ok's method and Schmid's method can get matches with higher completeness compared with our proposed method. However, from Table 5, we can clearly see that the efficiency of Ok's method is the worst. For Schmid's method, it also has more computational cost than our proposed method. In summary, our proposed method can quickly obtain many matches with completeness up to 50%. Thus, our proposed method can be used to process aerial stereo images with large size due to its efficiency.

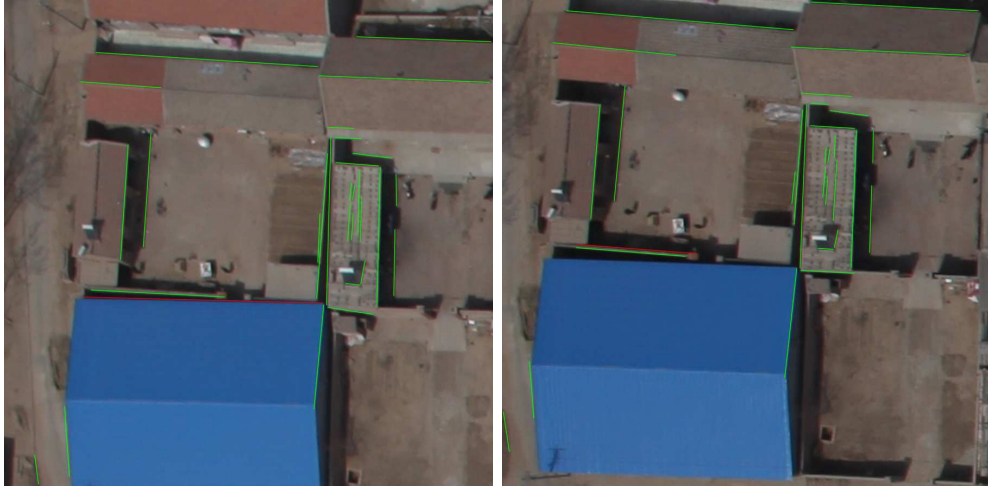


(a) Matched line segments of Aerial\_Large dataset obtained by the affine-invariant method (AILM). Total matches: 28. Correct matches: 22.

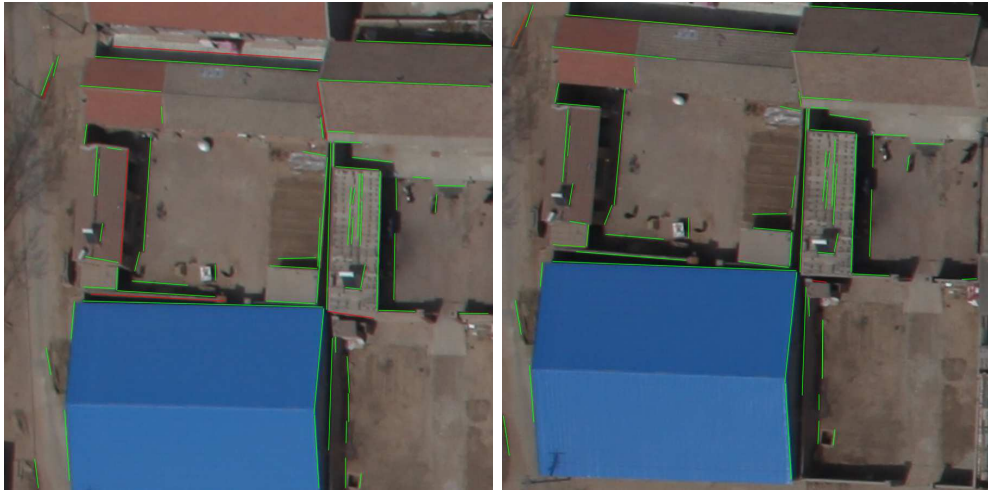


(b) Matched line segments of Aerial\_Large dataset obtained by Algorithm 1. Total matches: 56. Correct matches: 45.

Figure 14: Matching a pair of images in Aerial\_Large dataset using the affine-invariant method (AILM) and Algorithm 1.



(a) Matched line segments of DunHuan dataset obtained by the affine-invariant method (AILM). Total matches: 30. Correct matches: 29.



(b) Matched line segments of DunHuan dataset obtained by the proposed method. Total matches: 70. Correct matches: 64.

Figure 15: Line matching for a pair of images in Dunhuan dataset using the affine-invariant method (AILM) and Algorithm 1.

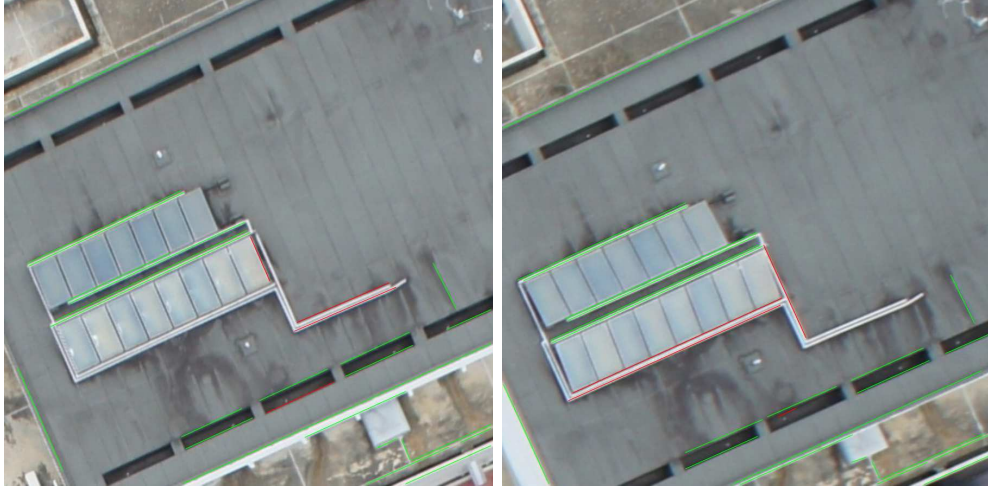


(a) Matched line segments of Tongzhou dataset obtained by the affine-invariant method (AILM). Total matches: 19. Correct matches: 18.



(b) Matched line segments of Tongzhou dataset obtained by the proposed method. Total matches: 82. Correct matches: 79.

Figure 16: Line matching for a pair of images in Tongzhou dataset using the affine-invariant method (AILM) and Algorithm 1.

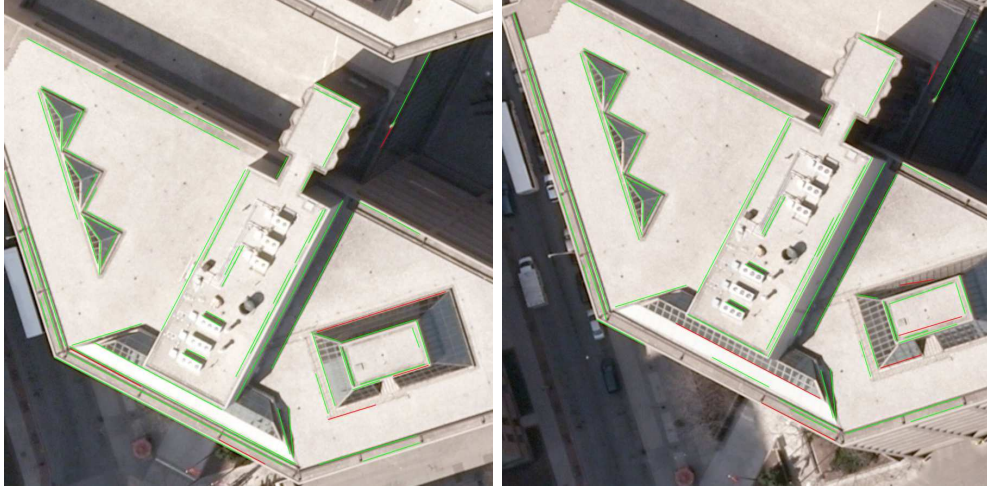


(a) Matched line segments of Guanzhou dataset obtained by the affine-invariant method (AILM). Total matches: 27. Correct matches: 23.

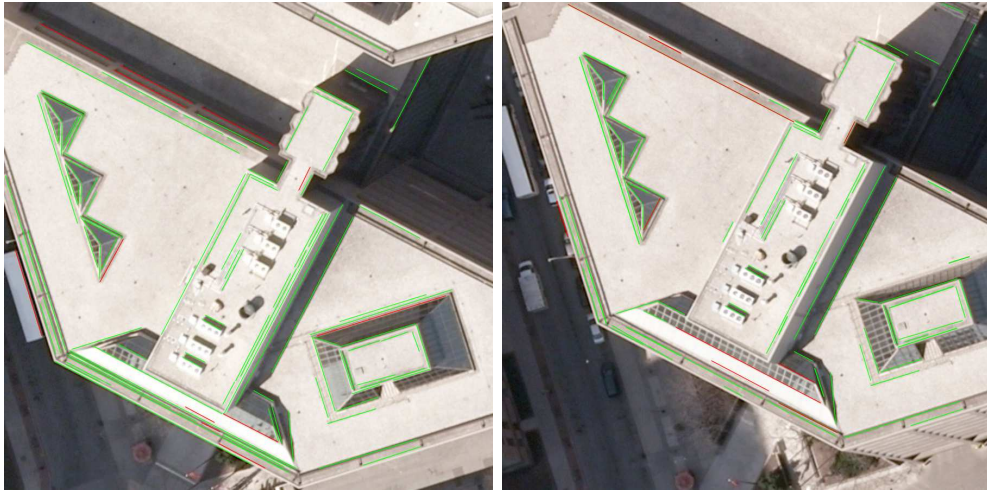


(b) Matched line segments of Guanzhou dataset obtained by the proposed method. Total matches: 84. Correct matches: 71.

Figure 17: Line matching for a pair of images in Guanzhou dataset using the affine-invariant method (AILM) and Algorithm 1.



(a) Matched line segments of Toronto dataset obtained by the affine-invariant method (AILM). Total matches: 76. Correct matches: 70.



(b) Matched line segments of Toronto dataset obtained by the proposed method. Total matches: 123. Correct matches: 110.

Figure 18: Line matching for a pair of images in Toronto dataset using the affine-invariant method (AILM) and Algorithm 1.

Secondly, the result of line matches of MSLD method is compared with our proposed method. Because MSLD method only supports its own line segment format, where a line segment is represented using not only two end-points but also line descriptors, we use line segment extraction function together with matching function of MSLD instead of LSD to generate unmatched line segments. To assure the same input, two end-points of the extracted line segments from MSLD are used in our proposed method. Here, DunHuan and testSite7 of Vaihingen datasets are used. From Table 6, the proposed method can obtain more correct matches with better efficiency. Especially for testSite7 dataset, MSLD has very poor performance. The main reason is that low-texture images cannot be processed by MSLD which extracts line descriptors based on grey-level information to guide matching.

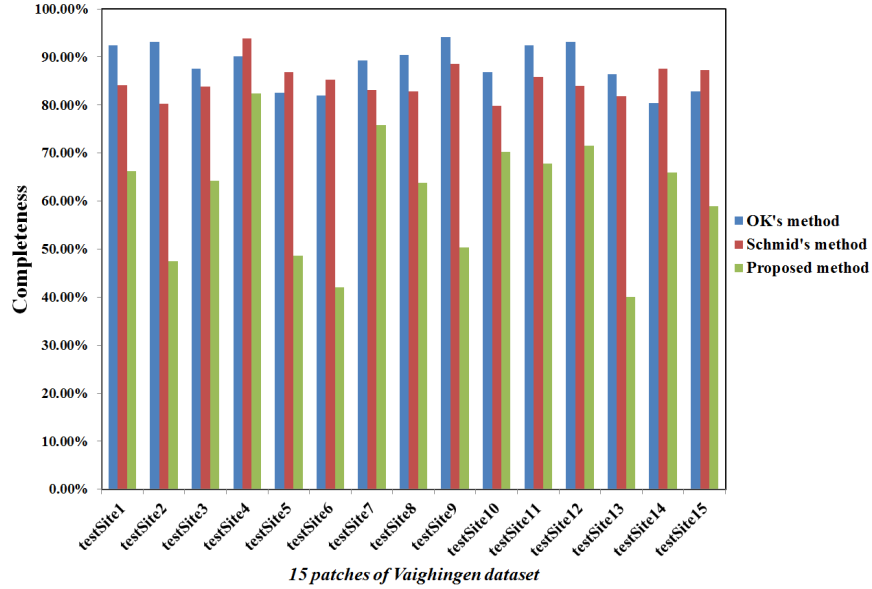


Figure 19: Using fifteen patches of Vaihingen dataset (Ok et al., 2012), we compare the completeness using OK's method, Schmid's method and the proposed method.

## 7. Conclusions and future work

We propose a line matching algorithm to deal with a pair of calibrated aerial images based on planar homography, using a mature point feature extraction and matching algorithm. The proposed algorithm matches line

	OK's method	Schmid's method	Proposed method
testSite 3	32.81	2.05	0.25
testSite 5	61.64	4.79	0.28
testSite 7	133.66	7.41	0.37
testSite 10	158.26	8.97	0.39

Table 5: Elapsed time (sec) for matching using OK’s method, Schmid’s method and the proposed method.

Parameter	DunHuan		testSite7	
	MSLD	Proposed method	MSLD	Proposed method
# Total Matches	13	15	37	65
# Wrong Matches	1	2	26	6
# Correctness	92.3%	86.6%	29.7%	90.7%
# Correct Matches	12	13	11	59
Completeness	54.5%	59.1%	15.7%	84.3%
Time (sec)	0.35	0.08	0.55	0.1

Table 6: Comparison of line matching between MSLD method and the proposed method.

segments using a fitted plane when three or more points provide an accurate planar homography and uses approximated planar homography when less than three points can be found. Using five pairs of aerial images, including rural and urban scenes, we showed that our proposed method is fast and can increase the number of matches with good quality, as compared with a method using 2D point feature correspondences (Fan et al., 2012). In addition, compared with OK’s and Schmid’s method (Ok et al., 2012) (Schmid and Zisserman, 1997), the proposed method has better efficiency.

The proposed method can be effective for most line segments. However, it may not be suitable for matching in cases such as eaves of building and street lamp, because the points on one side are not coplanar and the ordering constraint may fail for thin lines. In future work, we will investigate how to use the uncertainty of predicted line to choose the tentative line segments rather than using a given threshold. An additional aspect for future research will be to improve the number of matches and correctness by matching line segments using both 2D point correspondence and grey-level information.

## 8. Appendix

This appendix describes the derivation of a predicted line segment (Eq.(8) in Section 4).

The camera projection matrices of the source image and the target image are  $P_1 = K[I|0]$  and  $P_2 = K[R|t]$ . Given a line segment  $l_S = \{x_l^1, x_l^2\}$  on the source image, whose corresponding 3D line is parallel to a plane  $P_L = [n_L^T, d]^T$ , then, a predicted line segment  $l_E = \{x_E^1, x_E^2\}$  can be obtained by

$$\begin{aligned} l_E \propto x_E^2 \times x_E^1 &= (H^d x_l^2) \times (H^d x_l^1) \\ &= \det(H) H^{-T} (x_l^2 \times x_l^1) \\ &\propto (K(R - \frac{tn_L^T}{d})K^{-1})^{-T} (x_l^2 \times x_l^1) \\ &= (K^{-T}R^TK^T + (\frac{-K^{-T}n_L}{d})(Kt)^T)^{-1} (x_l^2 \times x_l^1). \end{aligned} \quad (10)$$

According to Sherman-Morrison (Sherman and Morrison, 1950),

$$(B + uv^T)^{-1} = B^{-1} + \frac{B^{-1}uv^TB^{-1}}{1 + v^TB^{-1}u} \quad (11)$$

where  $B$  is an invertible square matrix; and  $u, v$  are vectors. Thus, Eq.(10) can be written as

$$l_E \propto (K^{-T}RK^T + \frac{K^{-T}Rn_Lt^TRK^T}{d - n_L^TR^Tt})(x_l^2 \times x_l^1). \quad (12)$$

The distance from the origin of the source image to the plane on the world coordinate system can be expressed as

$$h = \frac{d}{\sqrt{n_L^T n_L}}. \quad (13)$$

So  $d = \pm h \sqrt{n_L^T n_L}$ . Substituted into Eq.(12), we have

$$l_E \propto (K^{-T}RK^T + \frac{K^{-T}Rn_Lt^TRK^T}{\pm h \sqrt{n_L^T n_L} - n_L^TR^Tt})(x_l^2 \times x_l^1). \quad (14)$$

## Acknowledgements

The authors would like to thank the authors (Werner, 2007), (Wang et al., 2009), (Fan et al., 2012), (Ok et al., 2012) for sharing their codes or test datasets, which were helpful to the research presented in this paper.

## References

- Bay, H., Ferrari, V., Van Gool, L., 2005. Wide-baseline stereo matching with line segments, in: Proc. Computer Vision and Pattern Recognition (CVPR), pp. 329–336.
- Chehata, N., Pierrot, M., Jung, F., Stamont, G., 2002. Extraction of 3D primitives from stereopairs of satellite images for automatic reconstruction of buildings, in: Proc. IAPR Workshop on Machine Vision Applications, pp.636–639.
- Cramer, M., 2010. The dgpf-test on digital airborne camera evaluation—overview and test design. Photogrammetrie-Fernerkundung-Geoinformation 2010, 73–82.
- Elaksher, A. F., 2011. Automatic line matching across multiple views based on geometric and radiometric properties. Applied Geomatics 3(1), 23–33.
- Engels, J., Arefi, H., Hahn, M., 2008. Generation of roof topologies using plane fitting with RANSAC. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS) 37.
- Fan, B., Wu, F., Hu, Z., 2012. Robust line matching through line–point invariants. Pattern Recognition 45, 794–805.
- Grompone von Gioi, R., Jakubowicz, J., Morel, J., Randall, G., 2012. Lsd: a line segment detector. Image Processing On Line.
- Habib, A. F., Zhai, R., Kim, C., 2010. Generation of complex polyhedral building models by integrating stereo-aerial imagery and lidar data. Photogrammetric Engineering and Remote Sensing, 776(5), 609–623.
- Hartley, R., 1995. A linear method for reconstruction from lines and points, in: Proc. Computer Vision (ICCV), pp. 882–887.

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9 Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision.  
10 Cambridge University Press.
- 11
- 12 Heuel, S., Forstner, W., 2001. Matching, reconstructing and grouping 3d  
13 lines from multiple views using uncertain projective geometry, in: Proc.  
14 Computer Vision and Pattern Recognition (CVPR), pp. 510–517.
- 15
- 16
- 17 Ke, Y., Sukthankar, R., 2004. Pca-sift: A more distinctive representation for  
18 local image descriptors, in: Proc. Computer Vision and Pattern Recognition  
19 (CVPR), pp. 506–513.
- 20
- 21
- 22 Long, T., Jiao, W., He, G., Wang, W., 2014. Automatic line segment regis-  
23 tration using gaussian mixture model and expectation-maximization algo-  
24 rithm. IEEE Journal of Selected Topics in Applied Earth Observations  
25 and Remote Sensing 7(5), 1688–1699.
- 26
- 27
- 28 Lourakis, M. I., Halkidis, S. T., Orphanoudakis, S. C., 2000. Matching dis-  
29 parate views of planar surfaces using projective invariants. Image and  
30 Vision Computing 18(9), 673–683.
- 31
- 32
- 33 Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints.  
34 International Journal of Computer Vision 60, 91–110.
- 35
- 36 Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local de-  
37 scriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence  
38 27, 1615–1630.
- 39
- 40 Novák, D., Baltasavias, E., Schindler, K., 2011. Reliable image matching with  
41 recursive tiling, in: Photogrammetric Image Analysis. Springer, pp. 49–60.
- 42
- 43 Ok, A. O., Wegner, J. D., Heipke, C., Rottensteiner, F., Soergel, U., Toprak,  
44 V., 2012. Matching of straight line segments from aerial stereo images of  
45 urban areas. ISPRS Journal of Photogrammetry and Remote Sensing, 74,  
46 133–152.
- 47
- 48
- 49 Ooi, B.C., McDonell, K.J., Sacks-Davis, R., 1987. Spatial kd-tree: An in-  
50 dexing mechanism for spatial databases, in: IEEE COMPSAC, p. 85.
- 51
- 52
- 53 Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J.D., 2013. Isprs test project  
54 on urban classification and 3d building reconstruction. Commission III-  
55 Photogrammetric Computer Vision and Image Analysis, Working Group  
56 III/4-3D Scene Analysis, 1–17.
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9 Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S.,
- 10 Breitkopf, U., 2012. The isprs benchmark on urban object classification
- 11 and 3d building reconstruction. Proc. XXII ISPRS Congress, Melbourne,
- 12 293–298.
- 13
- 14
- 15 Schindler, K., 2006. Geometry and construction of straight lines in log-polar
- 16 images. Computer Vision and Image Understanding 103, 196–207.
- 17
- 18 Schmid, C., Zisserman, A., 1997. Automatic line matching across views, in:
- 19 Proc. Computer Vision and Pattern Recognition (CVPR), pp. 666–671.
- 20
- 21
- 22 Sherman, J., Morrison, W.J., 1950. Adjustment of an inverse matrix cor-
- 23 responding to a change in one element of a given matrix. The Annals of
- 24 Mathematical Statistics, 124–127.
- 25
- 26 Stelmazyk, J.C.P., 1990. Measurement and integration of 3d structures by
- 27 tracking edges lines, in: Proc. European Conference on Computer Vision
- 28 (ECCV), pp. 269–280.
- 29
- 30
- 31 Sun, Y., Zhao, L., Huang, S., Yan, L., Dissanayake, G., 2014. L2-sift: Sift
- 32 feature extraction and matching for large images in large-scale aerial pho-
- 33 togrammetry. ISPRS Journal of Photogrammetry and Remote Sensing 91,
- 34 1–16.
- 35
- 36
- 37 Tang, A., Ng, T., Hung, Y., Leung, C.H., 2006. Projective reconstruction
- 38 from line-correspondences in multiple uncalibrated images. Pattern Recog-
- 39 nition 39, 889–896.
- 40
- 41
- 42 Wang, L., Neumann, U., You, S., 2009. Wide-baseline image matching us-
- 43 ing line signatures. Proc. International Conference on Computer Vision
- 44 (ECCV) pp.1311–1318.
- 45
- 46
- 47 Wang, Z., Wu, F., Hu, Z., 2009. Projective reconstruction from line cor-
- 48 respondences in multiple un-calibrated images. Pattern Recognition 42,
- 49 941–953.
- 50
- 51
- 52 Werner, T., 2007. Lmatch: Matlab toolbox for matching line segments ac-
- 53 cross multiple calibrated images.
- 54
- 55 Wu, C., 2007. Siftgpu: A gpu implementation of scale invariant feature
- 56 transform (sift). URL <http://cs.unc.edu/~ccwu/siftgpu>.
- 57
- 58

- 1  
2  
3  
4  
5  
6  
7  
8  
9 Zhang, L., Koch, R., 2012. Line matching using appearance similarities and  
10 geometric constraints. In Pattern Recognition, Lecture Notes in Computer  
11 Science.  
12  
13  
14 Zhang, L., 2005. Automatic digital surface model(DSM) generation from lin-  
15 ear array images. Mitteilungen-Institut für Geodäsie und Photogrammetrie  
16 an der Eidgenössischen Technischen Hochschule Zurich.  
17  
18 Zhao, L., Huang, S., Sun, Y., Yan, L., Dissanayake, G., 2014. Parallaxba:  
19 Bundle adjustment using parallax angle feature parametrization. Interna-  
20 tional Journal of Robotic Research (IJRR)(To Appear).  
21  
22  
23 Zhao, L., Huang, S., Yan, L., Dissanayake, G., 2011. Parallax angle  
24 parametrization for monocular slam, in: Proc. IEEE International Con-  
25 ference on Robotics and Automation (ICRA), pp. 3117–3124.  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65