

Article

An Accurate Linear Method for 3D Line Reconstruction for Binocular or Multiple View Stereo Vision

Lijun Zhong ¹, Junyou Qin ¹, Xia Yang ¹, Xiaohu Zhang ^{1,*}, Yang Shang ², Hongliang Zhang ² and Qifeng Yu ²¹ School of Aeronautics and Astronautics, Sun Yat-Sen University, Guangzhou 510275, China;

zhonglj9@mail.sysu.edu.cn (L.Z.); qin jy6@mail2.sysu.edu.cn (J.Q.); yangxia7@mail.sysu.edu.cn (X.Y.)

² College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410003, China; shangyang1977@nudt.edu.cn (Y.S.); hl_zhang_2000@hotmail.com (H.Z.); yuqifeng@vip.sina.com.cn (Q.Y.)

* Correspondence: zhangxiaohu@mail.sysu.edu.cn

Abstract: For the problem of 3D line reconstruction in binocular or multiple view stereo vision, when there are no corresponding points on the line, the method called Direction-then-Point (DtP) can be used, and if there are two pairs of corresponding points on the line, the method called Two Points 3D coordinates (TPS) can be used. However, when there is only one pair of corresponding points on the line, can we get the better accuracy than DtP for 3D line reconstruction? In this paper, a linear and more accurate method called Point-then-Direction (PtD) is proposed. First, we used the intersection method to obtain the 3D point's coordinate from its corresponding image points. Then, we used this point as a position on the line to calculate the direction of the line by minimizing the image angle residual. PtD is also suitable for multiple camera systems. The simulation results demonstrate that PtD increases the accuracy of both the direction and the position of the 3D line compared to DtP. At the same time, PtD achieves a better result in direction of the 3D line than TPS, but has a lower accuracy in the position of 3D lines than TPS.



Citation: Zhong, L.; Qin, J.; Yang, X.; Zhang, X.; Shang, Y.; Zhang, H.; Yu, Q. An Accurate Linear Method for 3D Line Reconstruction for Binocular or Multiple View Stereo Vision. *Sensors* **2021**, *21*, 658. <https://doi.org/10.3390/s21020658>

Received: 24 December 2020

Accepted: 14 January 2021

Published: 19 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: plane intersection; 3D line reconstruction; Point-then-Direction; corresponding image points intersection; linear solution

1. Introduction

The 3D line reconstruction is a basic problem in computer vision and optical measurements. It is widely used in computer vision problem, such as 3D scene reconstruction, non-cooperative target reconstruction and pose estimation for symmetric targets. Line features in the image are robust, stable, and easy to extract; the corresponding lines and points are often used in 3D reconstruction. In the position and attitude estimation for the non-cooperative target, the non-cooperative target is often reconstructed by points, lines, rectangles, or other shapes composed of lines, then the reconstructed model is used for tracking and relative pose estimation in subsequent frames. In pose estimation for the long symmetrical target, such as rockets and missiles, the direction of the target's central axis is often used to represent the target's attitude, which is often reconstructed by the Direction-then-Point (DtP) method. From the above analysis, we can determine that the accuracy of the 3D line is of significance for computer vision.

There are many studies on the reconstruction and measurement of non-cooperative targets [1–11] or attitude measurement of long symmetric targets based on binocular or multiple stereo vision. There are many kinds of non-cooperative targets to be reconstructed, especially targets in the space, such as satellites and spacecrafts. Many of them are based on feature points [2], point clouds [3,4], ellipses, circles [5,6] and line structures [7,8], and so on. However, the number of the feature points is huge, and the matches between them are more complicated and less robust than the feature lines. The ellipses and circles are more difficult to extract than the feature lines. There always are obvious line features on

the target. After the corresponding line matching, the reconstructed line can be used to reconstruct the target.

At present, there is not much research on 3D line reconstruction. For binocular stereo vision, the plane intersection method [12] and Two Points (TPS) are the main two methods for line reconstruction. The principle of the plane intersection method is that the 3D line locates on the plane consisting of the optical center and the line segment in the corresponding image. The intersection line of the planes of the two cameras can be calculated. We can obtain a point on the line; therefore, we also call this method Direction-then-Point. TPS is a direct method for 3D line construction. It utilizes two pairs of corresponding image points on the 2D line (generally, they are the endpoints), then two 3D points on the line can be calculated, and the 3D line is uniquely determined. Obviously, this method does not utilize the information of the 2D line, and its accuracy depends on the points' accuracy. DtP has a lower requirement for not needing corresponding points on the 2D line. Because the degeneracy of 3D lines is more severe than for 3D points, only 3D points on the baseline cannot be reconstructed, and the 3D line coplanar with the baseline cannot be reconstructed [13]; when the 3D line is close to coplanar with the baseline, the TPS can achieve higher accuracy than the DtP method. For a multiple camera system, the iterative method [13] based on minimizing the distance between the extracted line segment in the image and the line projected to the image is used. The reconstructed line is used by many other algorithms in computer vision. Many scholars analyze the influence of camera position, intersection angle, and relationship between the line and camera's imaging plane on the accuracy of the reconstruction result of the 3D line [11–20]. For 3D line reconstruction in stereo vision, the optimal condition is that the angle between the optical axes of the two cameras is 90 degrees, and the 3D line is parallel to both imaging planes of the two cameras and perpendicular to, not coplanar with, the baseline [18]. However, because the baseline of the stereo cameras is always short relative to the target distance, the requirement for two cameras with near-vertical optical axes cannot be satisfied. At the same time, because the 3D lines on the target always form a certain structure, such as a triangle, a quadrangle, etc., the requirement that every 3D line be nearly parallel to the imaging plane cannot be satisfied too.

As can be seen from the above description, DtP only uses line information and does not need corresponding points' information, but it has low position accuracy because it does not use the position information of the cameras. TPS uses two pairs of corresponding points; therefore, it can achieve better positional accuracy than DtP. However, it is difficult to obtain two pairs of corresponding image points under certain conditions. Considering the above situation, we proposed a novel method for line reconstruction based on a pair of corresponding image points and the 2D line direction. Firstly, we used the intersection method to obtain the 3D coordinates in the world coordinate system from the corresponding image points [21–23]. Secondly, we used this point as a point on the line to be solved to calculate the direction of the line. The point on the line is obtained before the line direction, thus call this method is named Point-then-Direction (PtD). The differences between the DtP, PtD and TPS methods are shown in Table 1.

Table 1. The differences between Direction-then-Point (DtP), Point-then-Direction (PtD) and Two Points (TPS) methods.

| Difference | DTP | PtD | TPS |
|---|-----|-----|-----|
| Pairs of corresponding image points | 0 | 1 | 2 |
| Can be reconstructed while the line is coplanar with the baseline (no point on the baseline) | × | × | ✓ |
| Can be reconstructed while the line is coplanar with the baseline (one point on the baseline) | × | × | × |
| Use of the camera's optical center | × | ✓ | ✓ |
| Use of the information of the 2D line | ✓ | ✓ | × |

The simulation results demonstrated that PtD can achieve more accurate results, both on direction and position than DtP, because it combines the information of the corresponding image points. At the same time, it can achieve a more accurate direction but less accurate position than TPS. If there is only one pair of corresponding image points on the line, the linear method proposed in this paper can achieve better results.

2. Direction-then-Point Method

2.1. Stereo-Vision Scenario

A pinhole camera model [12] was used in this study. As shown in Figure 1, The world coordinate system is $O_w - X_w Y_w Z_w$, which is a right-handed coordinate system. The image coordinate system is $O_i - X_i Y_i$, O_i in the top left corner of the image, X_i faces right, and Y_i faces down. The camera coordinate system is $O_c - X_c Y_c Z_c$, where O_c is the camera's optical center. The z -axis is pointing in the positive direction of the optical axis; the x -axis is perpendicular to the z -axis horizontally to the right; and the y -axis is perpendicular to the x -axis and to the z -axis, whose direction is determined by the right-handed definition criterion.

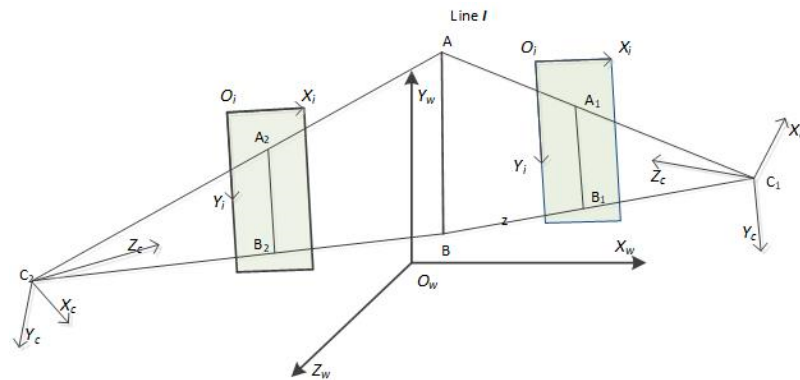


Figure 1. Plane intersection for stereo vision. AB is the 3D line I , C_1 and C_2 are the optical centers of the two cameras, A_1B_1 is the projection of I on camera C_1 , and A_2B_2 is the projection of I on camera C_2 .

2.2. Plane Representation by a Single Camera

As shown in Figure 1, AB is the 3D line I , C_1 and C_2 are the optical centers of the two cameras, A_1B_1 is the projection of I on camera C_1 , and A_2B_2 is the projection of I on camera C_2 . When atmospheric refraction is not considered, we can assume that I is on the plane $C_1A_1B_1$ and also on the plane $C_2A_2B_2$. Therefore, if we can attain both of the planes' equations in the world coordinate system, then we can obtain the direction of the 3D line and a point on the line.

Next, we derive its expression by taking a camera as an example. The equation of the line in the image coordinate system is:

$$A^T X = 0 \quad (1)$$

where $A = [a \ b \ c]^T$ and $X = [x \ y \ 1]^T$. The image plane is the $z_c = f$ plane in the camera coordinate system, where f is the physical focal length of the camera. The z -axis of the camera coordinate system passes through the principal point of the image plane (C_x, C_y) , so the transformation from the image coordinate system to the camera coordinate system is determined as:

$$\begin{cases} x_c = (x - C_x)d_x \\ y_c = (y - C_y)d_y \end{cases} \quad (2)$$

where (x_c, y_c) is in the camera coordinate system, (x, y) is in the image coordinate system, and (d_x, d_y) is the actual physical size of the camera pixel.

The line in the image can be expressed in the camera coordinate system as:

$$\begin{cases} a(\frac{x_c}{d_x} + C_x) + b(\frac{y_c}{d_y} + C_y) + c = 0 \\ z_c - f = 0 \end{cases} \quad (3)$$

The equivalent focal length of the camera is abbreviated as:

$$\begin{cases} F_x = f/d_x \\ F_y = f/d_y \end{cases} \quad (4)$$

The plane equation can be summarized as:

$$F_x x_c + b F_y y_c + (a C_x + b C_y + c) z_c = 0 \quad (5)$$

Given the parameter matrix $\mathbf{n} = [a F_x \quad b F_y \quad a C_x + b C_y + c]^T$, the plane equation is then rewritten as:

$$\mathbf{n}^T \mathbf{P}_c = 0 \quad (6)$$

where $\mathbf{P}_c = [x_c \quad y_c \quad z_c]^T$ is a point from the camera coordinate system, and \mathbf{n} is the normal vector of the plane. \mathbf{R} and \mathbf{t} are the rotation matrix and the translation vector from the world coordinate system to the camera coordinate system, respectively. Therefore, the point in the world coordinate system can be transformed to the camera coordinate system using Equation (7).

$$\mathbf{P}_c = \mathbf{R} \mathbf{P}_w + \mathbf{t} \quad (7)$$

where $\mathbf{P}_w = [x_w \quad y_w \quad z_w]^T$ is a point in the world coordinate system. The plane equation in the world coordinate system can then be derived as:

$$\mathbf{n}^T \mathbf{R} \mathbf{P}_w + \mathbf{n}^T \mathbf{t} = 0 \quad (8)$$

2.3. Plane Intersection

The plane equations of the two cameras, respectively, are:

$$\begin{cases} \mathbf{n}_1^T \mathbf{P}_w + d_1 = 0 \\ \mathbf{n}_2^T \mathbf{P}_w + d_2 = 0 \end{cases} \quad (9)$$

The 3D line $\mathbf{I} = (l, m, n)$ must be located on these two planes. Therefore, the direction vector of \mathbf{I} can be solved as:

$$\mathbf{I} = \mathbf{n}_1^T \times \mathbf{n}_2^T = \begin{bmatrix} i & j & k \\ \mathbf{n}_{11} & \mathbf{n}_{12} & \mathbf{n}_{13} \\ \mathbf{n}_{21} & \mathbf{n}_{22} & \mathbf{n}_{23} \end{bmatrix}. \quad (10)$$

A 3D line can be uniquely determined by a point and the direction vector. Therefore, we chose the foot point $P_0(x_0, y_0, z_0)$ from the O_w to the \mathbf{I} ; $(l, m, n) * P_0^T = 0$. P_0 can be solved by $(l, m, n) * P_0^T = 0$ and Equation (9). The point is directly solved by direction, and as a result the overall accuracy mainly depends on the accuracy of the line direction.

3. Point-then-Direction Method

3.1. Stereo Vision Scenario for PtD

As shown in Figure 1, Camera C_1 and Camera C_2 observe the line \mathbf{I} from different directions, and the projections on the image are $A_1 B_1$ and $A_2 B_2$, respectively. In this paper, we assume that one of the two endpoints of the line segments are the corresponding image points, i.e., A_1 and A_2 are the corresponding image points, or B_1 and B_2 are the corresponding image points. If the camera parameters and the line extracted from the image are error-free, the 3D point will be on the line obtained by the DtP, and then the

results obtained by the DtP and Ptd are same. However, if either the camera parameters or the line extracted from the image have an error, the 3D point may not be on the line I .

In a similar manner to the image-space residual method for camera calibration, the line I was projected to the image plane according to the theoretical model, and the angle between the projection and the 2D line extracted on the image plane was considered as the angle residual. We also minimized the square sum of all the angles. In this paper, this term is called the IAR. As shown in Figure 2, the $C_2A_2B_2$ plane is the plane composed of the optical center and the axis of the camera; $A'B'$ is the 3D line; B_2A_3 is the projection of $A'B'$ on the image plane; and α is the image angle residual.

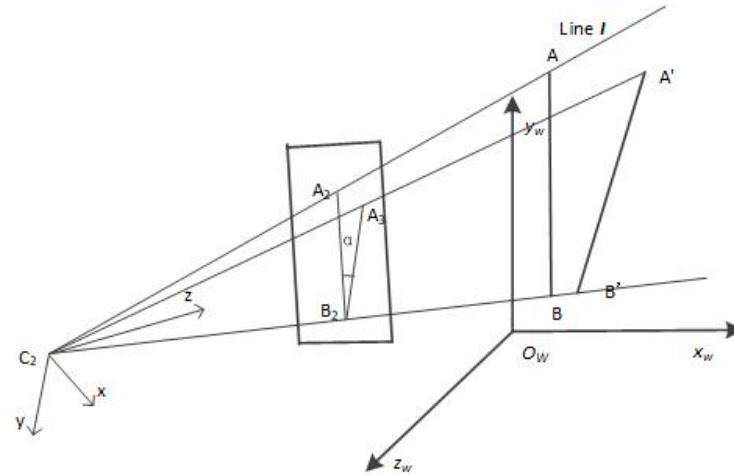


Figure 2. Schematic diagram of the image angle residual (IAR). The $C_2A_2B_2$ plane is the plane composed of the optical center and the axis of the camera; AB is the 3D line; $A'B'$ is the reconstructed result of the 3D line; B_2A_3 is the projection of $A'B'$ on the image plane; and α is the IAR.

From the above definition and stereo-vision scenario, we can ascertain that the 3D line passes through the 3D point, and the intersection of the plane composed of the 3D line and the camera's optical center and the corresponding imaging plane is parallel to the 2D line on the image. For a multiple camera system, we can obtain the least square solution.

The mathematical expression is derived below.

3.2. Image-Space Angle Residual

3.2.1. Definition of the Problem

It is assumed that the direction of the 3D line I is (l, m, n) , and I passes through a point $P_0(x_0, y_0, z_0)$ in the world coordinate system. It is easy to see that I also passes through the point $P_1(x_0 + l, y_0 + m, z_0 + n)$. In this paper, the P_0 is obtained by the intersection method of minimizing the distance from the spatial point to the back projection ray to obtain the three-dimensional coordinates from the corresponding image points A_1 and A_2 [24]. The plane of each camera passes through the point P_0 , P_1 and the corresponding optical center. The plane consisting of I and the optical center of the i th camera can be expressed as:

$$\begin{vmatrix} x & y & z & 1 \\ Ox_i & Oy_i & Oz_i & 1 \\ x_0 & y_0 & z_0 & 1 \\ x_0 + l & y_0 + m & z_0 + n & 1 \end{vmatrix} = 0 \quad (11)$$

Or

$$\begin{vmatrix} x & y & z & 1 \\ Ox_i & Oy_i & Oz_i & 1 \\ x_0 & y_0 & z_0 & 1 \\ l & m & n & 0 \end{vmatrix} = 0 \quad (12)$$

where (Ox_i, Oy_i, Oz_i) is the optical center of the i th camera. The plane equation after simplification is:

$$\mathbf{n}_{wi}^T \mathbf{P}_w + d_i = 0 \quad (13)$$

It can be seen from Equation (12) that \mathbf{n}_{wi} can be represented linearly by \mathbf{I} . Letting the relationship matrix be \mathbf{M} :

$$\mathbf{n}_{wi}^T = \mathbf{I}^T \mathbf{M}_i \quad (14)$$

It is obvious that \mathbf{M} is a full rank matrix. Assuming the transformation matrix from the world coordinate system to the camera coordinate system is $\begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ 0 & 1 \end{bmatrix}$, the plane is expressed in the camera coordinate system as:

$$\mathbf{n}_{ci}^T = \mathbf{n}_{wi}^T \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ 0 & 1 \end{bmatrix} \quad (15)$$

From the image plane equation $z = f$, the direction of the intersection line is $\mathbf{I}_i = (\mathbf{n}_{ci1}, \mathbf{n}_{ci2}, 0)$. Then, the angle between the 2D line in the image plane and the projection of \mathbf{I} on the corresponding image is:

$$\alpha_i = \cos^{-1} \left(\frac{[\mathbf{n}_{ci1} \quad \mathbf{n}_{ci2}] [\begin{smallmatrix} a_i & b_i \end{smallmatrix}]^T}{\|\mathbf{n}_{ci1} \quad \mathbf{n}_{ci2}\| \|\begin{smallmatrix} a_i & b_i \end{smallmatrix}\|} \right) \quad (16)$$

where a_i and b_i are coefficients of the linear equation $a_i x + b_i y + c_i = 0$ in the image plane.

Therefore, the optimized objective function is:

$$\min \sum_{i=1}^N \alpha_i^2 \quad (17)$$

where N is the number of cameras.

3.2.2. Linear Method

A method of linearly solving for the minimum value is given by the derivation below.

The original expression is very complicated, especially the partial derivative expression. The numerator and the denominator contain (l, m, n) , which cannot be solved linearly. Therefore, the expression is deformed.

If the two 2D lines $\mathbf{I}_1 = (a_1, b_1, c_1)^T, \mathbf{I}_2 = (a_2, b_2, c_2)^T$ are nearly parallel, the angle is:

$$\alpha_i = \tan^{-1} a_1 / b_1 - \tan^{-1} a_2 / b_2 \quad (18)$$

This is a small angle, and we have:

$$|\alpha_i| \approx |\tan(\alpha_i)| = \left| \frac{\frac{a_1}{b_1} - \frac{a_2}{b_2}}{1 + \frac{a_1 a_2}{b_1 b_2}} \right| = \left| \frac{a_1 b_2 - a_2 b_1}{a_1 a_2 + b_1 b_2} \right| \quad (19)$$

For the sake of discussion, we divided a_1, b_1 and a_2, b_2 by the larger one, respectively, so that the maximum value was 1. Then:

$$1 \leq |a_1 a_2 + b_1 b_2| \approx |a_1^2 + b_1^2| \leq 2 \quad (20)$$

where $|\alpha_i|$ is positively correlated with $|a_1 b_2 - a_2 b_1|$. $|\alpha|$ is the absolute value of α . Then, Equation (17) can be used instead of Equation (21) as the optimization solution objective function:

$$\min \left(\sum_{i=1}^N (a_i n_{ci2} - b_i n_{ci1})^2 \right) \quad (21)$$

From the previous derivation, $\mathbf{n}_{ci1}, \mathbf{n}_{ci2}$ is a linear function of $\mathbf{I} = (l, m, n)^T$. The equation can therefore be recorded as $\sum_{i=1}^N (\mathbf{G}_i^T \mathbf{I})^2$, where \mathbf{G}_i is the coefficient of \mathbf{I} in Equation (18), and it can be representative of a normal vector to a plane.

This can be thought of as a function of $(l, m, n)^T$, which is a function of (l, m, n) . Clearly, this function is a basic elementary function—it is continuous and differentiable in the domain of the definition, and the minimum value must exist. At the point where the minimum value is obtained, the partial derivative of (l, m, n) exists and it is 0, and the partial derivative is obtained separately.

$$\begin{cases} y'_l = \sum_{i=1}^N (\mathbf{G}_{i1} \mathbf{G}_i^T \mathbf{I}) = 0 \\ y'_m = \sum_{i=1}^N (\mathbf{G}_{i2} \mathbf{G}_i^T \mathbf{I}) = 0 \\ y'_n = \sum_{i=1}^N (\mathbf{G}_{i3} \mathbf{G}_i^T \mathbf{I}) = 0 \end{cases} \quad (22)$$

We can solve Equation (19) by the SVD [25] decomposition of the coefficient matrix to obtain (l, m, n) .

3.2.3. Efficiency Analysis

The linear method for minimizing the image angle residual is as follows.

1. Solve the 3D coordinate of the corresponding image points with a time complexity of $O(n)$;
2. Solve the line parameters projected onto the image plane with a time complexity of $O(n)$;
3. Solve the coefficient matrix with a time complexity of $O(n)$;
4. Find the final result in fixed time.

It can be seen from the above analysis that the image angle residual method can be solved only by calculating the 3D point coordinate, the projection line parameter, the solution coefficient matrix, and the least-squares solution, so the time complexity is $O(n)$.

4. Experiments

Because the ground truth was not easy to obtain in the real experiment, we used the simulation to verify the accuracy of the algorithm and then we used only the real data to verify the validity of the algorithm.

4.1. Simulation

We simulated many conditions to validate the accuracy of our method. Firstly, the variation of the intersection angle of the two cameras and the 3D line's attitude variation were simulated to verify the accuracy and robustness of the proposed algorithm. Secondly, the various error conditions for the stereo vision were simulated, including the 2D line extraction error, the external camera parameter error, and both the above errors. Thirdly, we verified the accuracy while in a multiple camera system. Finally, we assessed the time performance of our method in a multiple camera system.

4.1.1. Simulation Environment

The simulation platform was Windows 10 Pro, the implementation language of the algorithm was C++, the implementation environment was Microsoft Visual Studio 2013, and the processor was an Intel(R) Core TM i7-9850H 2.6GHz.

The equivalent focal length was (2181.8, 2181.8) and the image principal point was (1024, 1024). The cameras were arranged in a circle around the target, and the radius of the circle was 4.5 m. The origin of the world coordinate system was the center of the circle. The two endpoints of the 3D line were $A(-0.5, -0.5, -0.5)$ and $B(1.5, 1.5, 1.5)$. The top view of the simulation scenario is shown in Figure 3.

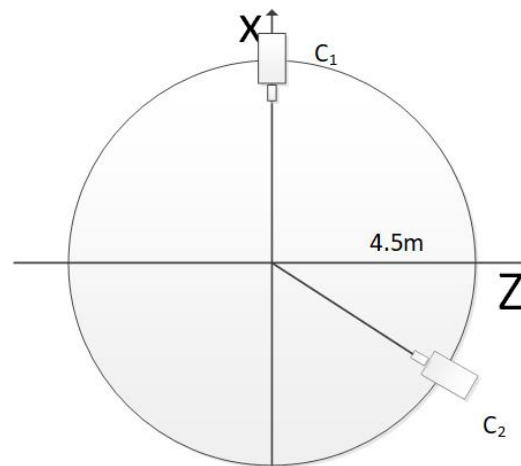


Figure 3. The top view of the simulation scenario.

4.1.2. Simulation Description

1. The corresponding image points used in the PtD method were obtained by adding the same error to the projection of the endpoint A in the image, which ensured that PtD method was also solved under the same error conditions.
2. The TPS, DtP, and PtD methods were used for each simulation. We used the angle between the result and the ground truth and the distance from the two endpoints to the result line to evaluate the accuracy. All simulations were calculated 1000 times, and the root mean square (RMS) of the angular error and the average of distance were obtained. The units of angular error were degrees, and the units of distance error were meters.

$$E = \cos^{-1} \frac{\mathbf{I}_t}{\|\mathbf{I}\| \|\mathbf{I}_t\|} \quad (23)$$

$$E_d = \text{abs}(d_{AI}) + \text{abs}(d_{BI}) \quad (24)$$

In the above equation $\mathbf{I}_t(l_t, m_t, n_t)$ is the ground truth, and $\mathbf{I}(l, m, n)$ is the algorithm's result. d_{AI} is the distance between the endpoint A of the 3D line segment to the result line, and d_{BI} is the distance between the endpoint B of the 3D line segment to the result line.

Simulations were conducted separately to determine the following conditions:

1. Angle between optical axes of binocular cameras.
2. 3D line's attitude.
3. Line segments extraction error in image.
4. Errors of all camera external parameters.
5. All camera external parameters and extraction errors.
6. Number of cameras.
7. Running time.

Case 1 was to verify the precision under the different angle between the two cameras. Case 2 was to verify the precision under the target's different attitude. Cases 3–5 were to verify the precision under the error of the cameras' parameters. Case 6 was to verify the precision as the number of cameras increased. Case 7 was to test the temporal performance of the algorithm.

4.1.3. Simulation Results

The simulation details and results are given, respectively.

Firstly, we simulated the intersection angles of two cameras, which varied to verify the accuracy of the three methods. As shown in Figure 3, C_1 was placed in the fixed point (4.5,0,0), and C_2 was moved on the circle with a radius of 4.5 m. The angle between the optical axes of C_1 and C_2 varied from 30° to 150° . We simulated two conditions of the 3D line, one of which was the 3D line perpendicular to the baseline of C_1 and C_2 . When

the 3D line was perpendicular to the baseline, the endpoints of the 3D line segment were $A(-0.5, -0.5, -0.5)$ and $B(-0.5, 1.5, -0.5)$. From the definition of the simulation scenario, we knew that, as the intersection angle changed from 30° to 150° , the line segment AB was always perpendicular to the baseline, and the distance between AB and the baseline became smaller and smaller. When the 3D line is not perpendicular to the baseline, the endpoints of the 3D line segment were $A(-0.5, -0.5, -0.5)$, $B(1.5, 1.5, 1.5)$. A Gaussian error with a mean of 0 was added to the optical center of the camera, with a 5 cm RMS value. A Gaussian error with a mean of 0 was added to the camera angle, with a 1° RMS value. A Gaussian error with a mean of 0 was added to the x - and y -directions of the two endpoints of the 2D line segments, with a 2-pixel RMS value. The results are shown in Figure 4.

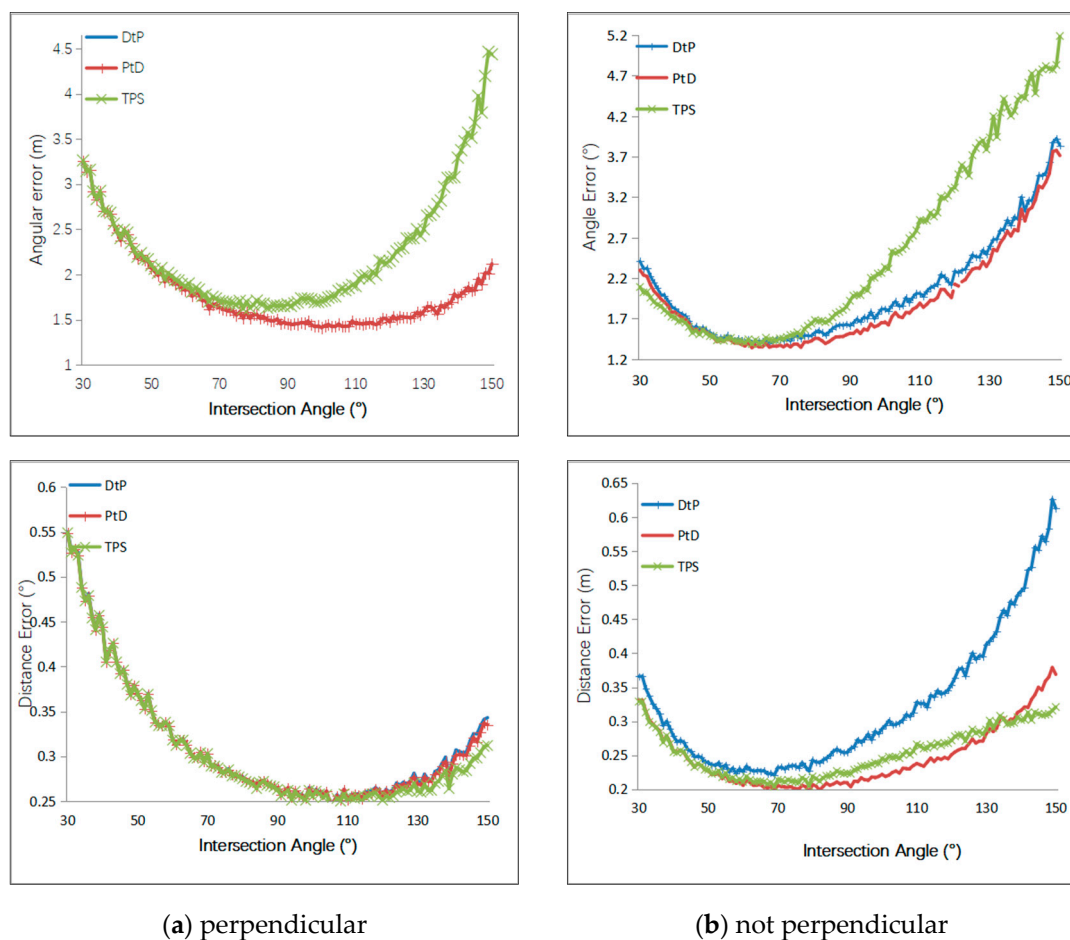


Figure 4. Relationship between the angular and positioning error and the intersection angle. (a) The result when the 3D line was parallel to the imaging plane and perpendicular to the baseline; and (b) the result when the 3D line was not parallel to the imaging plane and was not perpendicular to the baseline.

When the 3D line remained unchanged, the error was relevant to the intersection angle. In the beginning, the three methods achieved the same accuracy. As the intersection angle varied from 30° to 150° , the distance between the baseline and AB grew smaller and smaller. In other words, they were becoming closer and closer to each other, which meant the intersection condition was getting worse; the PtD method had the same accuracy as the DtP method. TPS had the best positional accuracy but the worst angular accuracy. In the condition where the 3D line was not parallel to the imaging plane and was not perpendicular to the baseline, PtD had higher accuracy both on the position and angle than DtP; TPS achieved the worst angular accuracy, and DtP achieved the worst positional accuracy.

Secondly, we simulated the condition where the 3D line moved while the intersection angle remained unchanged; the C_1 and C_2 places were fixed. Endpoint A was $(-0.5, -0.5, -0.5)$, The initial position of endpoint B was $(-0.5, 1.5, -0.5)$, and the k th position was $(-0.5 - k * 0.05, 1.5, -0.5 - k * 0.05)$; k was the offset time. The end position of point B was $(-5.5, 1.5, -5.5)$. A Gaussian error with a mean of 0 was added to the optical center of the camera, with a 5 cm RMS value. A Gaussian error with a mean of 0 was added to the camera angle, with a 1° RMS value. A Gaussian error with a mean of 0 was added to the x - and y -directions of the two endpoints of the 2D line, with a 2-pixel RMS value. The results of the simulation while the intersection angles are 90° and 120° , and shown in Figure 5a,b respectively.

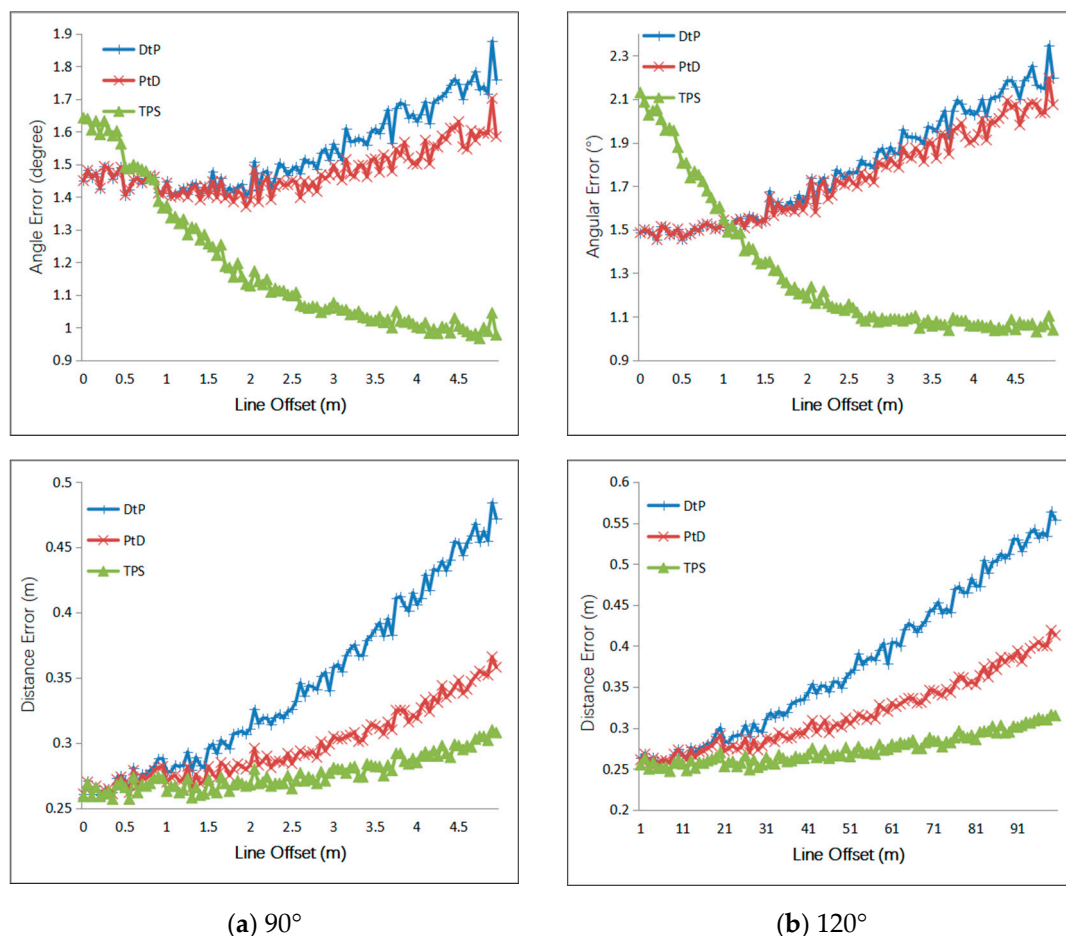


Figure 5. Relationship between the angular and positioning error and the line offset. (a) The result when the intersection angle was 90° ; and (b) the result when the intersection angle was 120° .

As shown in Figure 5, at the start, the TPS method had the best positioning accuracy and the worst angular accuracy. However, as the intersection condition became, the TPS had the best accuracy both in direction and position. The reasons are as follows.

Firstly, the 3D line became longer, and the final length was 3.68-times greater than that of the initial length. Therefore, theoretically, if only the influence of the line length was considered, the direction error of TPS would be the $1/3.68$ of the original value. However, the error of the 3D point B increased because the intersection angle of 3D point B became smaller. Therefore, the error reduction was not proportional.

Secondly, the angle between the 3D line and the baseline remained unchanged—it was always 90° . However, the angle between the 3D line and the imaging plane varied from 0 to 45° and the distance between the 3D line and the baseline became smaller. At the start, when the intersection angle was 90° , the distance between the 3D line and

the baseline was 3.89 m; when the intersection angle was 120° , the distance between the 3D line and the baseline was 2.94 m. At the end, however, the distances were 2.50 m and 2.30 m, respectively. The lengths of the baseline were 6.36 m and 7.79 m, respectively. From the above analysis, we can determine that the smaller distance aggravated the degeneracy of the lines' reconstruction.

Thirdly, we simulated the condition where the cameras and the 3D line remained unchanged, but the errors of the camera's external parameters and 2D line were different. The intersection angle was 120° . The endpoints of the 3D line were $A(0.5, -0.5, 0.5)$ and $B(1.5, 1.5, 1.5)$. We simulated for 2D line error, the error of the camera external parameters, and all above errors. The error details were as below:

Case 1: A Gaussian error with a mean of 0 was added to the x - and y -directions at the head and the tail of the line segment, with 0–4 pixels of RMS. The results are shown in Figure 6.

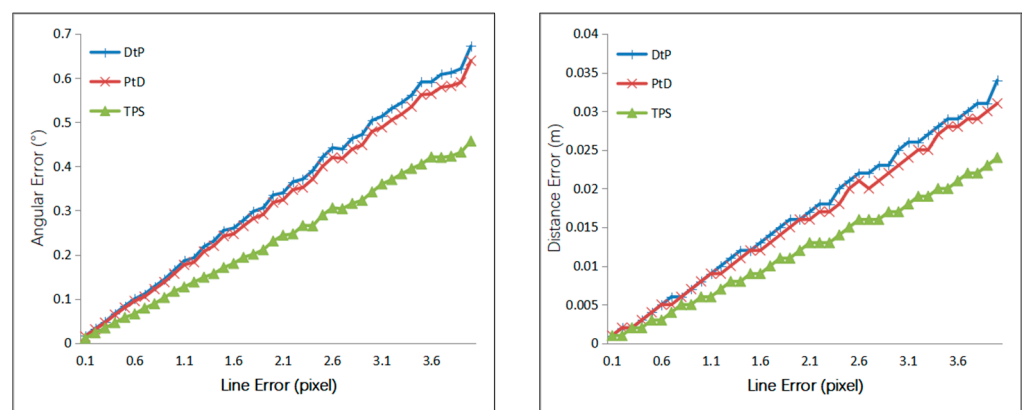


Figure 6. Relationship between the line error and error of 2D line.

From Figure 6, we can see that if only the same 2D line error existed, the 3D line's accuracy of the three methods, from most to least accurate, was TPS, PtD, then DtP.

Case 2: A Gaussian error with a mean of 0 was added to the optical center of the camera, with 1 mm to 50 mm of RMS, and the step size was 1 mm. A Gaussian error with a mean of 0 was added to the camera angle, with 0.05 – 2.5° of RMS. The simulation results are shown in Figure 7.

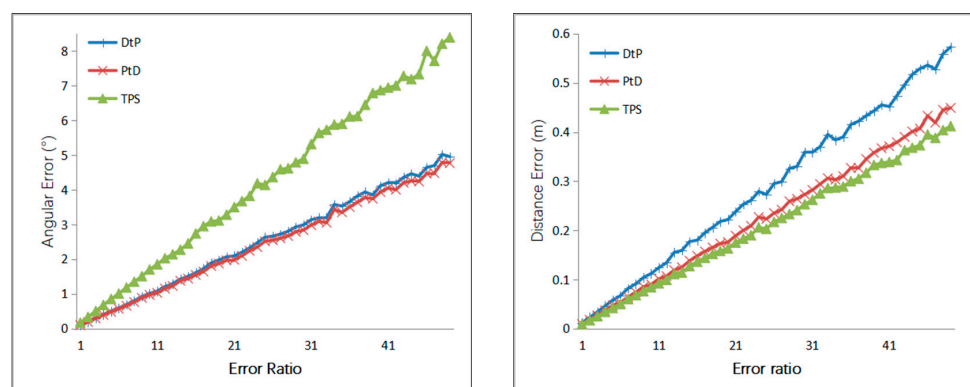


Figure 7. Relationship between the line error and error of external camera parameters.

From Figure 7, we can see that under the same error of the camera's external parameters, the directional accuracy of the 3D line of the three methods, from most to least accurate, was PtD, DtP, then TPS, and the positional accuracy of the 3D line of the three methods from most to least accurate, was TPS, PtD, then DtP.

Case 3: A Gaussian error with a mean of 0 was added to the optical center of the camera, with 1 mm to 50 mm of RMS, and the step size was 1 mm. A Gaussian error with a mean of 0 was added to the camera angle, with 0.05–2.5° of RMS. A Gaussian error with a mean of 0 was added to the x - and y -directions of the two endpoints of the 3D line, with 0.1–5 pixels of RMS. The simulation results are shown in Figure 8.

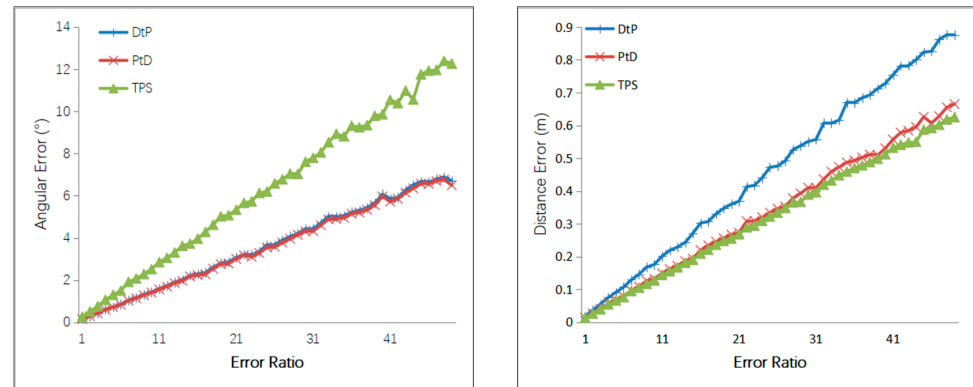


Figure 8. Relationship between the line error and the error of intersection parameters.

From Figure 8, we can see that under same error of external camera's parameters and the 2D lines, the 3D line's direction accuracy of the three methods, from most to least accurate, was PtD, DtP, then TPS, and the 3D line's position accuracy of the three methods, from most to least accurate, was TPS, PtD, then DtP. The position accuracies of TPS and PtD are close, and the directional accuracy of PtD was far better than TPS.

Additionally, we simulated the condition where the camera number varied from 2 to 20; the cameras were evenly distributed on the circle as shown in Figure 3. The DtP method used the LS [26] directly solve this scenario. A Gaussian error with a mean of 0 was added to the optical center of the camera, with a 10 mm root mean square (RMS) value. A Gaussian error with a mean of 0 was added to the camera angle, with a 0.5° RMS value. A Gaussian error with a mean of 0 was added to the x - y -directions of the two endpoints of the 3D line, with a 1-pixel RMS value. The endpoints of the 3D line were $A(0.5, -0.5, 0.5)$, $B(1.5, 1.5, 1.5)$. The results are shown in Figure 9.

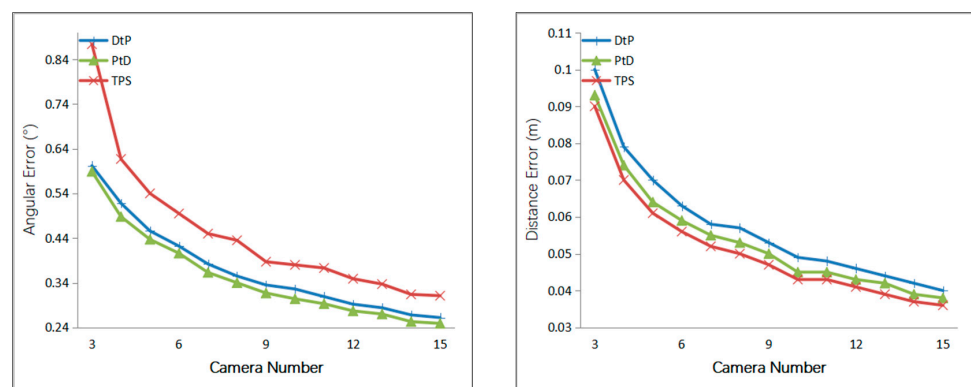


Figure 9. Relationship between the line error and the camera number.

From Figure 9, we can see that as the number of cameras increased, the accuracy of the three methods increased gradually, indicating that these methods utilized the constraints of the multi-camera. Similar to the result in Figure 8, with the increase in the number of cameras, under the same error of the external camera's parameters and the 2D lines, the 3D line's direction accuracy of the three methods, from most to least accurate, was PtD, DtP, then TPS, and the 3D line's positional accuracy of the three methods, from most to least

accurate, was TPS, PtD, then DtP. The positional accuracies of TPS and PtD were close, and the directional accuracy of the PtD method was far better than the TPS method.

Finally, we simulated the condition where the camera number increased from 2 to 20 to verify the time performance of our method. The linear algorithm took a small amount of time; therefore, for the convenience of comparison, we counted the time it took to run 10,000 times. A Gaussian error with a mean of 0 was added to the optical center of the camera, with a 5 cm RMS value. A Gaussian error with a mean of 0 was added to the camera angle, with a 1° RMS value. A Gaussian error with a mean of 0 was added to the x - y -directions of the two endpoints of the 2D line segments, with a 2-pixel RMS value. The simulation results are shown in Figure 10.

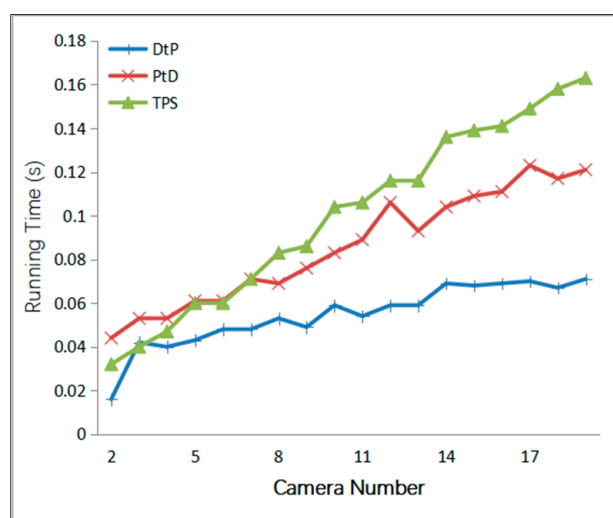


Figure 10. Relationship between the running time (in seconds) and the number of cameras. TPS, PtD, and DtP are linear methods, but PtD takes longer than DtP.

Figure 10 confirms that all three methods are linear methods; TPS took the longest, then PtD, and the fastest was DtP. DtP only needs to solve each plane equation and then solve the least-squares method; PtD needs to solve the 3D point coordinate and the line parameters in two steps; TPS needs to solve the 3D point coordinate two times, which explains the variances in running time between the methods.

4.1.4. Simulation Conclusion

The simulation results are shown in Table 2. The “F” in the table means first, the “S” means second, and the “T” means third and the “S->T” means from second to third.

Table 2. All simulation results.

| Method | Case 1 | | Case 2 | | Case 3 | Case 4 | Case 5 | Case 6 |
|--------|---------------|-------------------|--------|------|--------|--------|--------|--------|
| | Perpendicular | Not Perpendicular | 90° | 120° | | | | |
| (A)DtP | S | F | S→T | S→T | T | S | S | S |
| PtD | F | F | F→S | F→S | S | F | F | F |
| TPS | T | S | T→F | T→F | F | T | T | T |
| DtP(D) | T | S | T | T | T | T | T | T |
| PtD | S | S | S | S | S | S | S | S |
| TPS | F | F | F | F | F | F | F | F |

From the above simulation results, we can come to some conclusions:

1. Under the same error of the external camera parameters and the 2D line, the PtD method has the best directional accuracy, and the TPS method has the best positional accuracy.
2. When the intersection condition was normal, the 3D line's direction accuracy of the three methods, from most to least accurate, was PtD, DtP, then TPS, and the 3D line's position accuracy of the three methods, from most to least accurate, was TPS, PtD, and DtP. Therefore, if the direction accuracy is more important, the PtD method should be used, and if positional accuracy is more important, the TPS method should be used—provided that there are two pairs of corresponding image points.
3. When the intersection condition was bad and the target was relative long, TPS achieved the best accuracy, both in the direction and position of the 3D line. However, it needs two pairs of corresponding image points, which is sometimes hard to satisfy. PtD achieved better accuracy than DtP both in the direction and position of the 3D line, so PtD can be used if there is only one pair of corresponding image points.

4.2. Physical Experiment

We used physical experiment 1 to verify the precision of our method, and used physical experiment 2 (in which TPS could not be used) to verify the correctness of the algorithm.

4.2.1. Physical Experiment 1

Physical Environment

We used three cameras to measure the lines—the camera setup is shown in Figure 11. The physical environment is shown in Figure 12. The target was a checkerboard and was placed in front of the cameras. The precision of the checkerboard was 0.01 mm. The 3D lines to be reconstructed were the four edge lines in the checkerboard, as marked in Figure 12. The length of the lines 1, 3, 5, and 7 were 30 cm, and the lines 2, 4, 6, 8 were 21 cm. From the positional relationship between the camera and the target, we could ascertain that (a) was in a better intersection condition than (b), and the lines 1 and 3 were in a better intersection condition than the lines 2 and 4. All cameras were calibrated by the diagonal markers behind the checkerboard, and the 3D coordinates of the diagonal markers were obtained by the total station Leica TS60, with an accuracy of 0.6 mm. At the same time, we also obtained the 3D coordinates of the four corners of the checkerboard using the Leica TS60. PtD, DtP, and TPS methods were used to reconstruct the four 3D lines. The camera and lens models were Basler (acA1440—220uc) and RICOH (6 mm 1:1.4), respectively.

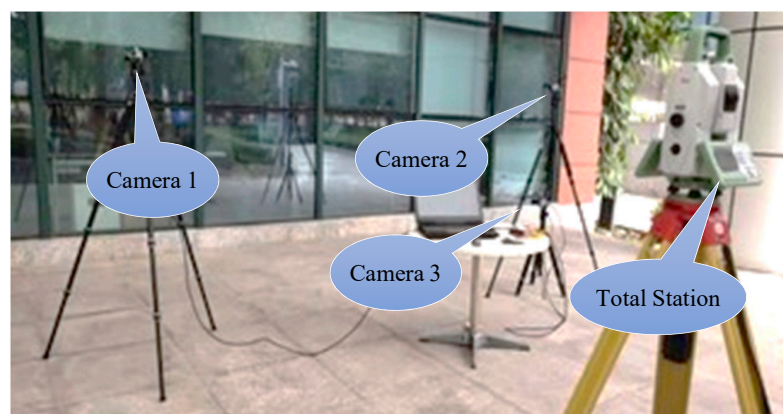


Figure 11. Physical scene for experiment 1.



Figure 12. The image captured by camera 1. The eight 3D lines to be reconstructed are marked by the red lines on the images; (a) is more perpendicular and in a better intersection condition than (b).

Experimental Procedure

The procedure of the experiment was as follows:

1. The world coordinate system was established by the total station. The total station was used to obtain the coordinates of the calibration control points. The coordinate system direction was vertically upwards (Y) and horizontal (XZ), and the system constituted a right-hand coordinate system.
2. The stereo cameras were used to take photos synchronously. The checkerboard was placed at two angles to verify the accuracy of the methods in two conditions. One was where the 3D lines were parallel to the imaging plane, and the other was not. The calibration control points were extracted from the image, and the 3D coordinates were used to calibrate all of the cameras in a unified world coordinate system. All parameters of the camera were calibrated, including the main point, equivalent focal length, lens distortion, and external parameters of the camera.
3. The 2D lines and the corresponding image points were extracted from the two images by extracting the two endpoints of the 2D lines. The corresponding image points were one of the endpoints of the 2D lines.
4. After the correction of lens distortion, DtP, PtD, and TPS were used separately to obtain the 3D line *I*.

First, we estimated the precision by the distance of two points—the results are shown in Table 3. From the results, we determined that the maximum of the distance error was 0.2 mm and the corresponding angle was $\tan^{-1}\left(\frac{0.2}{300}\right) = 0.038^\circ$. The theoretical accuracy of the total station was 0.6 mm, which meant a corresponding angle $\tan^{-1}\left(\frac{0.6}{210}\right) = 0.16^\circ$.

Table 3. Errors of the total station (mm).

| Line No. | Measured Distance | Ideal Distance |
|----------|-------------------|----------------|
| 1 | 300.0 | 300.0 |
| 2 | 210.1 | 210.0 |
| 3 | 300.2 | 300.0 |
| 4 | 210.0 | 210.0 |
| 5 | 300.1 | 300.0 |
| 6 | 210.1 | 210.0 |
| 7 | 300.0 | 300.0 |
| 8 | 210.0 | 210.0 |

The errors between the coordinates obtained by the total station and given by the intersection method [21] to demonstrate the precision of the cameras' parameters are shown in Table 4. The i th point was the top or left endpoint of the i th line.

Table 4. Results of 3D points' errors (mm).

| Point No. | X | Y | Z |
|-----------|------|------|------|
| 1 | −0.2 | −0.7 | −0.0 |
| 2 | −0.5 | −0.5 | −0.1 |
| 3 | −0.2 | −0.6 | −0.0 |
| 4 | 0.4 | −0.5 | 0.8 |
| 5 | −0.4 | −0.2 | 0.6 |
| 6 | −0.0 | −0.0 | 0.1 |
| 7 | −0.5 | −0.6 | 0.2 |
| 8 | −0.1 | −0.5 | 0.7 |

Therefore, we used the point coordinates obtained by the total station to calculate the direction of the line as the ground truth I_t . We calculated the angle between I_t and I as the angular error, and the sum of distances from the two endpoints to I as the distance error, to estimate the accuracy of our method.

All the results are shown in Tables 5 and 6. The second, third and fourth columns are angular errors ($^{\circ}$), and the fifth, sixth and seventh columns are distance errors (mm).

Table 5. Experiment results for good conditions.

| No. | DtP | PtD | TPS | DtP | PtD | TPS |
|-----|-------------|-------------|-------------|-----|------------|------------|
| 1 | 0.10 | 0.10 | 0.09 | 2.4 | 1.2 | 1.4 |
| 3 | 0.10 | 0.11 | 0.12 | 1.6 | 1.5 | 1.4 |
| 5 | 0.12 | 0.12 | 0.12 | 0.9 | 0.8 | 0.7 |
| 7 | 0.06 | 0.07 | 0.08 | 1.6 | 1.5 | 1.6 |
| sum | 0.38 | 0.4 | 0.41 | 6.5 | 5 | 5.1 |

Table 6. Experiment results for bad conditions.

| No. | DtP | PtD | TPS | DtP | PtD | TPS |
|-----|-------------|-------------|-------------|------|------------|------------|
| 2 | 0.58 | 0.28 | 0.03 | 7.7 | 2.0 | 1.1 |
| 4 | 0.78 | 0.50 | 0.23 | 6.8 | 2.0 | 1.6 |
| 6 | 0.07 | 0.07 | 0.16 | 2.9 | 0.4 | 0.7 |
| 8 | 1.07 | 1.08 | 0.08 | 4.1 | 4.3 | 1.5 |
| sum | 2.5 | 1.93 | 0.5 | 21.5 | 8.7 | 4.9 |

From the above tables, we can see that for the lines 1, 3, 5, and 7, the sum of the angular errors of the 3D line, from lowest to highest, was TPS, PtD, then DtP, and they were very close; the maximum difference was 0.02. The sum of the distance error of the 3D line, from highest to lowest, was DtP, PtD, then TPS. For the lines 2, 4, 6, and 8, the accuracy of the direction and distance, from lowest to highest, was TPS, PtD, then DtP. However, the TPS needed two pairs of corresponding image points, which was difficult to meet under certain conditions, such as in physical experiment 2.

4.2.2. Physical Experiment 2

Physical experiment 2 was designed to verify the correctness of our method if there were more than two cameras.

Physical Environment

The setup for the physical experiment is shown in Figure 13. The target was placed on the board in the center, and the diagonal markers on the side were the calibration control

points. The 3D line to be reconstructed was the target's central axis. The resolution of the camera was 4288×2848 pixels, and the equivalent focal distance was approximately 5200.

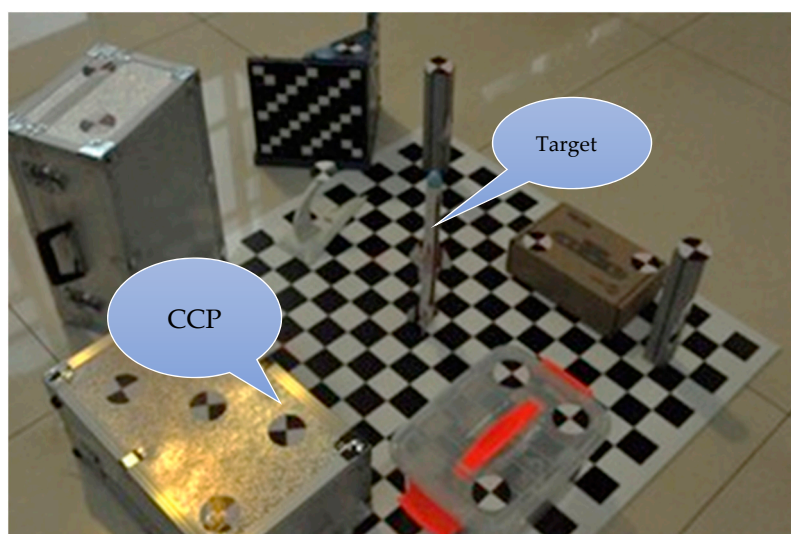


Figure 13. Physical scene for experiment 2. The target was placed on the board in the center, and the diagonal markers on the side were the calibration control points (CCPs). The 3D line to be reconstructed was the target's central axis.

Experimental Procedure

The experiment was conducted as follows:

1. The total station was used to obtain the coordinates of the calibration control points in the world coordinate system. The total station coordinate system was used as the world coordinate system. The coordinate system direction was vertically upwards (Y) and horizontal (XZ) and the system constituted a right-hand coordinate system.
2. The same camera was used to take photographs from nine different locations. The calibration control points were extracted from the image, and the three-dimensional coordinates were used to calibrate all of the cameras in a unified world coordinate system.
3. The 2D line was extracted from the image. The corresponding image point was the top point of the target.
4. DtP and PtD were used separately to obtain the results. For hard-to-find pairs of corresponding image points on the target, TPS was not used.

All the target directional results are shown in Table 7. It can be seen from the table that the results of the two methods were essentially the same.

Table 7. Experimental results.

| Camera No. | DtP | PtD |
|------------|---------------------|---------------------|
| 2 | (−0.012, 1, −0.014) | (−0.012, 1, −0.014) |
| 6 | (−0.020, 1, −0.022) | (−0.016, 1, −0.019) |
| 9 | (−0.017, 1, −0.012) | (−0.018, 1, −0.014) |

5. Conclusions

Considering the problem of 3D line reconstruction in binocular or multiple view stereo vision, where there is only a pair of corresponding points on the line, a new method called PtD was proposed. Compared to DtP and TPS methods, PtD uses both point and line information for 3D line reconstruction. The simulation results in this paper demonstrate that our method achieves the best accuracy for determining direction; and although the positional accuracy of our method is worse than TPS, it is better than DtP. The physical

experiment shows that under bad intersection conditions, the PtD method achieves better accuracy than the DtP method.

Author Contributions: Q.Y., X.Z., Y.S., X.Y., H.Z., and L.Z. participated the design of the method; L.Z. carried out the software implementation, experiments design, data analysis and manuscript writing; X.Y. checked the manuscript; J.Q. carried out the physical experiments and data analysis; X.Z. provided financial support; All authors have read and approved the final manuscript.

Funding: This research was funded by [the National Natural Science Foundation of China] grant number [U1601651].

Informed Consent Statement: Not applicable.

Data Availability Statement: All experimental data in this paper were created by this study.

Conflicts of Interest: There is no conflict of interest regarding this article.

References

1. Liu, L.; Zhao, Z. A new approach for measurement of pitch, roll and yaw angles based on a circular feature. *Trans. Inst. Meas. Control* **2013**, *35*, 384–397. [\[CrossRef\]](#)
2. Hao, W.; Zhang, X.; Huang, Y.; Yang, F.; Guo, B. Determining relative position and attitude of a close non-cooperative target based on the SIFT algorithm. *J. Beijing Inst. Technol.* **2014**, *3*, 110–114.
3. Yin, F.; Chou, W.; Wu, Y.; Yang, G.; Xu, S. Sparse unorganized point cloud based relative pose estimation for uncooperative space target. *Sensors* **2018**, *18*, 1009. [\[CrossRef\]](#) [\[PubMed\]](#)
4. He, Y.; Liang, B.; He, J.; Li, S. Non-cooperative spacecraft pose tracking based on point cloud feature. *Acta Astronaut.* **2017**, *139*, 213–221. [\[CrossRef\]](#)
5. Peng, J.; Xu, W.; Yuan, H. An efficient pose measurement method of a space non-cooperative target based on stereo vision. *IEEE Access* **2017**, *5*, 22344–22362. [\[CrossRef\]](#)
6. Peng, J.; Xu, W.; Liang, B.; Wu, A.G. Pose measurement and motion estimation of space non-cooperative targets based on laser radar and stereo-vision fusion. *IEEE Sens. J.* **2018**, *19*, 3008–3019. [\[CrossRef\]](#)
7. Yu, F.; He, Z.; Qiao, B.; Yu, X. Stereo-vision-based relative pose estimation for the rendezvous and docking of noncooperative satellites. *Math. Probl. Eng.* **2014**, *2014*, 1–12. [\[CrossRef\]](#)
8. Du, X.; Liang, B.; Xu, W.; Qiu, Y. Pose measurement of large non-cooperative satellite based on collaborative cameras. *Acta Astronaut.* **2011**, *68*, 2047–2065. [\[CrossRef\]](#)
9. Miao, X.; Zhu, F.; Hao, Y. Pose estimation of non-cooperative spacecraft based on collaboration of space-ground and rectangle feature. In Proceedings of the International Symposium on Photoelectronic Detection and Imaging 2011: Space Exploration Technologies and Applications, Beijing, China, 24–26 May 2011; Volume 8196.
10. Kannan, S.K.; Johnson, E.N.; Watanabe, Y.; Sattigeri, R. Vision-based tracking of uncooperative targets. *Int. J. Aerosp. Eng.* **2011**, *2011*, 1–17. [\[CrossRef\]](#)
11. Baillard, C.; Schmid, C.; Zisserman, A.; Fitzgibbon, A. Automatic line matching and 3D reconstruction of buildings from multiple views. In Proceedings of the ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, Munich Germany, 8–10 September 1999; Volume 32, pp. 69–80.
12. Yu, Q.; Sun, X.; Chen, G. A new method of measure the pitching and yaw of the axes symmetry object through the optical image. *J. Natl. Univ. Def. Technol.* **2000**, *22*, 15–19.
13. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
14. Ok, A.Ö.; Wegner, J.D.; Heipke, C.; Rottensteiner, F.; Soergel, U.; Toprak, V. Accurate reconstruction of near-epipolar line segments from stereo aerial images. *Photogramm. Fernerkund. Geoinf.* **2012**, *2012*, 345–358. [\[CrossRef\]](#)
15. Ding, S. *Research on Axis Extraction Methods for Target's Attitude Measurement*; National University of Defense Technology: Changsha, China, 2013.
16. Zhang, Y.; Wang ZH, Q.; Qiao, Y.F. Attitude measurement method research for missile launch. *Chin. Opt.* **2015**, *8*, 997–1003. [\[CrossRef\]](#)
17. Luo, K.; Fan, L.; Gao, Y.; Zhang, H.; Li, Q.A.; Zhu, Y. Measuring technology on elevation angle and yawing angle of space target based on optical measurement method. *J. Changchun Univ. Sci. Technol.* **2007**, *30*, 12–14.
18. Wang, C. *Study on Methods to Improve the Attitude Measurement Precision of the Rotary Object*; University of Chinese Academy of Sciences (Institute of Optics and Electronics): Chengdu, China, 2013.
19. Chen, M.; Tang, Y.; Zou, X.; Huang, K.; Huang, Z.; Zhou, H.; Wang, C.; Lian, G. Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology. *Comput. Electron. Agric.* **2020**, *174*, 105508. [\[CrossRef\]](#)
20. Tang, Y.C.; Wang, C.; Luo, L.; Zou, X. Recognition and localization methods for vision-based fruit picking robots: A review. *Front. Plant Sci.* **2020**, *11*, 510. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Li, Y.; Bo, Y.; Zhao, G. Survey of measurement of position and pose for space non-cooperative target. In Proceedings of the IEEE 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 5101–5106.

-
22. Pesce, V.; Lavagna, M.; Bevilacqua, R. Stereovision-based pose and inertia estimation of unknown and uncooperative space objects. *Adv. Space Res.* **2017**, *59*, 236–251. [[CrossRef](#)]
 23. Opromolla, R.; Fasano, G.; Rufino, G.; Grassi, M. A model-based 3D template matching technique for pose acquisition of an uncooperative space object. *Sensors* **2015**, *15*, 6360–6382. [[CrossRef](#)]
 24. Yuan, Y. *Research on Networked Videometrics for the Shape and Deformation of Large-Scale Structure*; National University of Defense Technology: Changsha, China, 2013.
 25. Golub, G.H. Singular value decomposition and least squares solutions. *Numerische Mathematik* **1970**, *14*, 403–420. [[CrossRef](#)]
 26. Madsen, K.; Nielsen, H.B.; Tingleff, O. *Methods for Non-Linear Least Squares Problems*, 2nd ed.; Informatics and Mathematical Modelling, Technical University of Denmark: Lyngby, Denmark, 2004.