


Article

Pose and Focal Length Estimation Using Two Vanishing Points with Known Camera Position

Kai Guo * , Rui Cao *, Ye Tian, Binyuan Ji, Xuefeng Dong and Xuyang Li

Northwest Institute of Nuclear Technology, Xi'an 710024, China; ty_2122022@163.com (Y.T.); jibinyuan163@163.com (B.J.); dongxuefeng15@163.com (X.D.); lixuyang0827@163.com (X.L.)

* Correspondence: guokai@mail.nwpu.edu.cn (K.G.); b5403002@163.com (R.C.); Tel.: +86-185-102-37517 (K.G.)

Abstract: This paper proposes a new pose and focal length estimation method using two vanishing points and a known camera position. A vanishing point can determine the unit direction vector of the corresponding parallel lines in the camera frame, and as input, the unit direction vector of the corresponding parallel lines in the world frame is also known. Hence, the two units of direction vectors in camera and world frames, respectively, can be transformed into each other only through the rotation matrix that contains all the information of the camera pose. Then, two transformations can be obtained because there are two vanishing points. The two transformations of the unit direction vectors can be regarded as transformations of 3D points whose coordinates are the values of the corresponding unit direction vectors. The key point in this paper is that our problem with vanishing points is converted to rigid body transformation with 3D–3D point correspondences, which is the usual form in the PnP (perspective-n-point) problem. Additionally, this point simplifies our problem of pose estimation. In addition, in the camera frame, the camera position and two vanishing points can form two lines, respectively, and the angle between the two lines is equal to the angle between the corresponding two sets of parallel lines in the world frame. When using this geometric constraint, the focal length can be estimated quickly. The solutions of pose and focal length are both unique. The experiments show that our proposed method has good performances in numerical stability, noise sensitivity and computational speed with synthetic data and real scenarios and also has strong robustness to camera position noise.



Citation: Guo, K.; Cao, R.; Tian, Y.; Ji, B.; Dong, X.; Li, X. Pose and Focal Length Estimation Using Two Vanishing Points with Known Camera Position. *Sensors* **2023**, *23*, 3694. <https://doi.org/10.3390/s23073694>

Academic Editor: Luke Wallace

Received: 3 February 2023

Revised: 19 March 2023

Accepted: 30 March 2023

Published: 3 April 2023



Copyright: © 2023 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: pose estimation; vanishing point; focal length; camera position; unit direction vector

1. Introduction

The camera pose estimation, using accurate inputs, is an old but still widely studied topic. The accurate inputs are mainly points, lines and planes. If the relative pose needs to be estimated, the 2D–2D point or line correspondences are used, and many algorithms have been proposed [1–5]. If the absolute pose needs to be estimated, the 2D–3D point or line correspondences are used, and the corresponding methods are called PnP (perspective-n-point) solvers [6–9] and PnL (perspective-n-line) solvers [10–13]. In addition, there is a category of methods between the point-based and line-based correspondences, such as the pose estimation methods using vanishing points [14,15]. The 3D parallel lines in space will intersect at a point in the image plane called the vanishing point. When these methods estimate the pose, the 3D parallel lines and 2D vanishing points will be used. The method proposed in this paper is based on the vanishing points to estimate the focal length and pose. It also needs accurate inputs, which are similar to the PnP and PnL solvers. Hence, we briefly introduce the two categories of methods as follows.

When there is no other prior knowledge, the minimum point set for solving the PnP problem is three, and the corresponding methods are called P3P (perspective-three-point) solvers [16–19], which have a maximum of four solutions, showing the phenomenon of multiple solutions. One more constraint must be given in order to obtain the unique

solution, and these solvers cannot solve the intrinsic parameters, such as the focal length or principal point. Hence, some works of literature [20,21] proposed that the focal length could be estimated simultaneously as the pose estimation, and the minimum number of point correspondence sets is four. These methods solved the problem that the camera lens is unknown or the zoom lens is used. When the short focal lens or fisheye lens is used, image distortion exists. In this case, the distortion should be estimated to improve the accuracy. The literature from [22] proposed a method to estimate the pose, distortion and focal length simultaneously by using five 2D–3D point correspondences, called P5Pfr. However, it should be noted that the distortion here refers to the radial distortion—most of the distortion is radial distortion. Although the pose only contains six DOF (degrees of freedom), each degree of freedom contains trigonometric functions and is coupled to each other. If the minimum point set is used for estimation, the computational process is nonlinear and computational complexity is high. However, when the number of points used is not less than six, the pose estimation can be directly solved linearly [23–25], and the corresponding method is called DLT (direct linear transform). As the number of points used increases, the number of estimable parameters increases or the computational complexity decreases from a nonlinear to a linear solution. Of course, the difficulty of obtaining more accurate points also increases. Therefore, the choice of estimation method depends on the number of accurate points that can be obtained in the FOV (field of view) and whether the partial intrinsic parameters need to be estimated. In order to reduce complexity and improve precision, some parameters of the pose measured by the sensors can be used as prior knowledge. For example, the IMU (inertial measurement unit) is used to obtain a vertical direction [26–29], or RTK (real-time kinematic) is used to obtain camera positions [30–32]. These methods can reduce the number of required point correspondences while the number of estimated parameters is unchanged, and the accuracy and calculation speed are both improved.

Similar to the PnP problem, the minimum line set required to solve the PnL problem is also three, which is called the P3L (perspective-3-line) method [33,34]. The difference is that these methods have up to eight solutions, the computational complexity is higher, and the accuracy and computational speed are both decreased. More parameters can be estimated by using more 2D–3D line correspondences, reducing the computational complexity. When the number of 2D–3D line correspondences used is not less than six, it can be directly solved linearly [35]. Simultaneously, the computational complexity changes from the nonlinear solution of the P3L methods to a linear solution, which can improve the computational speed and accuracy. In addition, some methods use sensors to measure the partial pose information in advance, such as vertical direction [36,37] or camera position [38], to reduce the number of 2D–3D line correspondences required and improve the accuracy and computational speed. Furthermore, without reducing the number of 2D–3D line correspondences, some intrinsic parameters, such as focal length, can be simultaneously estimated.

There is a category of methods between the point-based and line-based correspondences, such as the pose estimation methods using vanishing points [39,40]. These methods use 2D vanishing point-3D parallel line correspondences, and our proposed method in this paper belongs to this category. One vanishing point can provide two constraints; hence, when there is no other prior knowledge, at least three vanishing points are needed to estimate the pose, and at least four vanishing points are needed if the focal length is estimated simultaneously. Grammatikopoulos [41] used two vanishing points to estimate the camera pose when two parallel line sets are perpendicular to each other and when the origin of the world frame is the intersection. Guo [42] used a single vanishing point to complete this job, but the disadvantage is that one orientation needs to be measured in advance.

In this paper, we propose a new method for estimating the focal length and pose by using two vanishing points and the camera's position. In the camera frame, the camera's position and each vanishing point can determine two lines. Additionally, according to the

definition of vanishing point, the angle between the two lines, which is a function of the focal length, is equal to the angle between the corresponding two parallel line sets, which are known in the world frame. Hence, an equation can be obtained with an unknown parameter, i.e., the focal length. Then, the focal length is estimated efficiently. In addition, the vanishing point can determine the unit direction vector of the corresponding parallel lines in the camera frame after the focal length estimation. Because the unit direction vector in the world frame is the input, it is known. Then, the transformation between the two unit direction vectors in the camera frame and the world frame, respectively, can be obtained using the rotation matrix between the camera frame and the world frame. There are two vanishing points; hence, two transformations can be obtained. The transformation of the unit direction vectors can be regarded as the transformation of 3D points; that is, a 3D point in a camera frame, whose coordinates are equal to the values of the corresponding unit direction vector in the camera frame, can be transformed to a 3D point in a world frame whose origin is located at the origin of the camera frame, whose coordinates are equal to the values of the corresponding unit direction vector in the world frame. This is the key point to estimate the pose in this paper. The experimental results show our proposed method performs well in terms of numerical stability, noise sensitivity and computational speed in synthetic data and real images.

The remainder of the paper is organized as follows. In Section 2, we provide the derivation of the focal length and pose estimation; Section 3 provides the experiments and results to show how well our method performs; Sections 4 and 5 are the discussion and conclusions, respectively.

2. Proposed Method

This paper provides two sets L_i ($i = 1,2$) of parallel lines with known direction vectors and a camera position O_c in the world frame $S_{w1} (O_w_X_w Y_w Z_w)$ to estimate the pose and focal length. To simplify the derivation, we assume each set has two 3D lines, namely L_{i-j} ($j = 1,2$), and then the corresponding projections of them on the image plane are denoted as l_{i-j} . The geometric construction is illustrated in Figure 1.

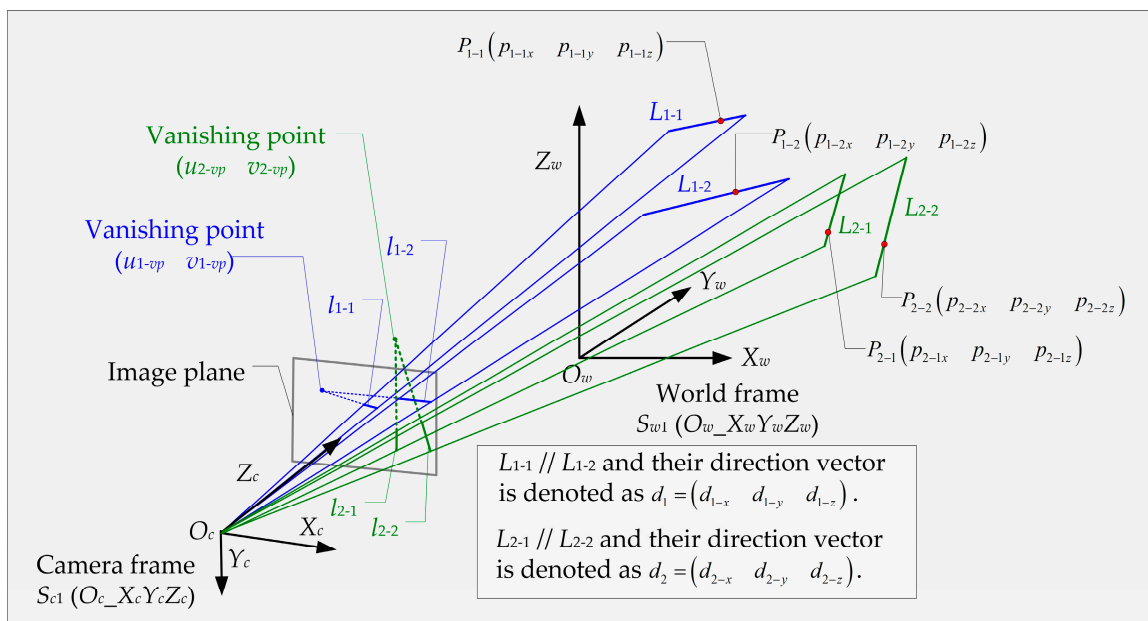


Figure 1. The geometric construction of this paper includes two sets of parallel lines (blue and green), the corresponding projections and vanishing points on the image plane.

Next, we will use two steps to estimate the pose and focal length, respectively.

2.1. Focal Length Estimation

In the camera frame S_{c1} ($O_c X_c Y_c Z_c$), the unit direction vector of the 3D line L_{i-j} is denoted as $d_i = (d_{i-x} \ d_{i-y} \ d_{i-z})$, which is unknown. Additionally, a 3D point $P_{i-j}(p_{i-jx} \ p_{i-jy} \ p_{i-jz})$, which is also unknown, is on the 3D line L_{i-j} . Now, the line L_{i-j} can be written as

$$L_{i-j} = P_{i-j} + k_{i-j} \cdot d_i \tag{1}$$

Here, k_{i-j} is an arbitrary scale factor. The vanishing point on the image plane is the projection of the 3D point located at the infinity spatial place. Here, we assume these 3D points can be seen and denoted as P_{v1}, P_{v2} in this paper. According to Equation (1), their coordinates can be written as

$$\begin{aligned} P_{v1} &= k_{v1} \cdot d_1, & k_{v1} &\rightarrow \infty \\ P_{v2} &= k_{v2} \cdot d_2, & k_{v2} &\rightarrow \infty \end{aligned} \tag{2}$$

where k_{v1}, k_{v2} are the scale factors of the 3D points located at the infinity spatial place. Their projections on the image plane are denoted as $p_{v1}(u_{1-vp} \ v_{1-vp}), p_{v2}(u_{2-vp} \ v_{2-vp})$. The geometric construction is illustrated in Figure 2.

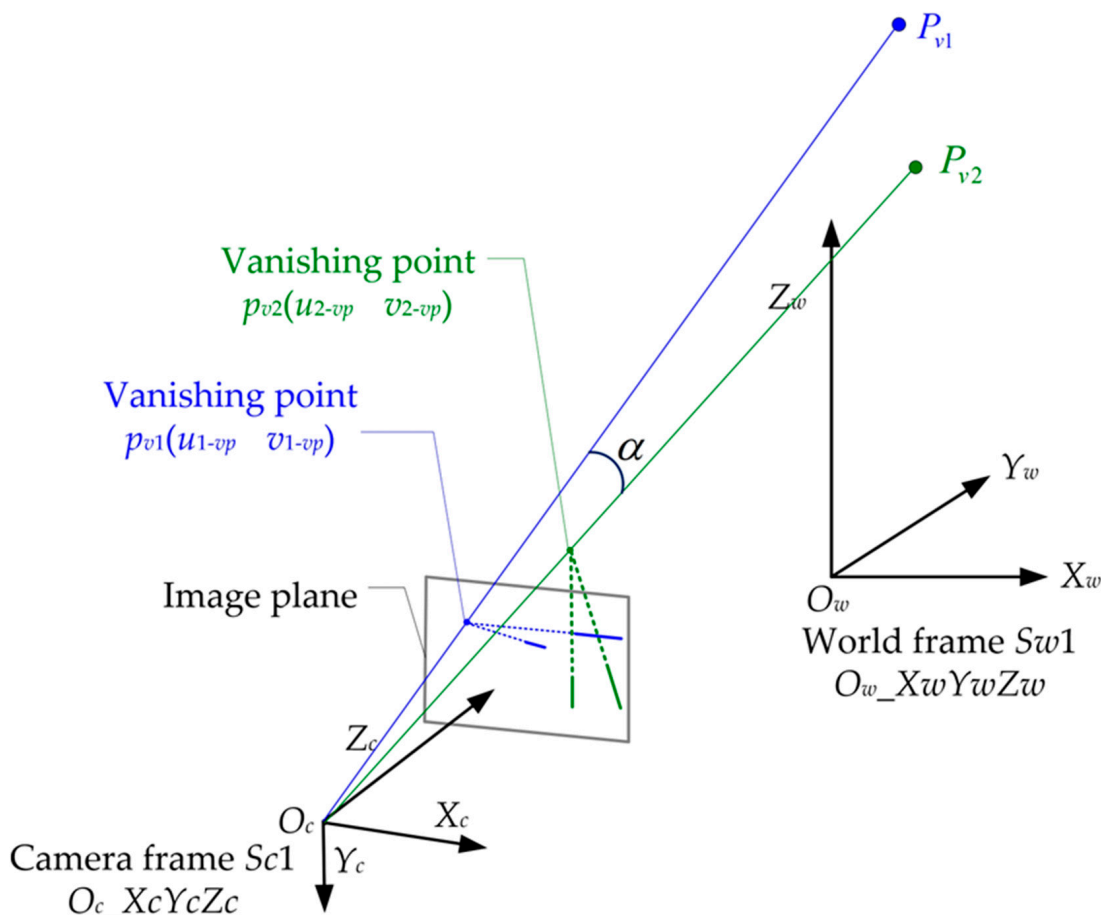


Figure 2. Two vanishing points were used for focal length estimation.

In Figure 2, α is not only the angle between the lines $O_c P_{v1}$ and $O_c P_{v2}$ in the world frame S_{w1} but is also the angle between the lines $O_c p_{v1}$ and $O_c p_{v2}$ in the camera frame S_{c1} . The expressions of the unit direction vectors of the lines $O_c P_{v1}$ and $O_c P_{v2}$ in the world frame S_{w1} can be written as

$$\begin{aligned} d_{v1} &= \lim_{k_{v1} \rightarrow \infty} \frac{k_{v1} d_1 - O_c}{\|k_{v1} d_1 - O_c\|} = d_1 \\ d_{v2} &= \lim_{k_{v2} \rightarrow \infty} \frac{k_{v2} d_2 - O_c}{\|k_{v2} d_2 - O_c\|} = d_2 \end{aligned} \tag{3}$$

Then, we can calculate the angle α using

$$\cos \alpha = \frac{d_{v1} \cdot d_{v2}}{\|d_{v1}\| \cdot \|d_{v2}\|} \tag{4}$$

In the camera frame S_{c1} , the expressions of the direction vectors of the lines $O_c p_{v1}$ and $O_c p_{v2}$ can be written as

$$\begin{aligned} O_c p_{v1} &= \begin{pmatrix} u_{1-vp} & v_{1-vp} & f \end{pmatrix} \\ O_c p_{v2} &= \begin{pmatrix} u_{2-vp} & v_{2-vp} & f \end{pmatrix} \end{aligned} \tag{5}$$

Here, f is the focal length in pixels. According to the characteristic of angle α , we can obtain

$$\cos \alpha = \frac{O_c p_{v1} \cdot O_c p_{v2}}{\|O_c p_{v1}\| \cdot \|O_c p_{v2}\|} = \frac{u_{1-vp} \cdot u_{2-vp} + v_{1-vp} \cdot v_{2-vp} + f^2}{\sqrt{u_{1-vp}^2 + v_{1-vp}^2 + f^2} \cdot \sqrt{u_{2-vp}^2 + v_{2-vp}^2 + f^2}} \tag{6}$$

Let $\cos \alpha = m_1$, $u_{1-vp} \cdot u_{2-vp} + v_{1-vp} \cdot v_{2-vp} = m_2$, $u_{1-vp}^2 + v_{1-vp}^2 = m_3$ and $u_{2-vp}^2 + v_{2-vp}^2 = m_4$. Then, we can simplify Equation (6) as

$$(m_1^2 - 1) \cdot f^4 + (m_1^2 m_3 + m_1^2 m_4 - 2m_2) \cdot f^2 + m_1^2 m_3 m_4 - m_2^2 = 0 \tag{7}$$

Here, f^2 is regarded as the unknown parameter, and the equation is a quadratic equation with one unknown. Two solutions of f^2 exist. Since $f > 0$ and $f^2 > 0$, we can obtain a unique solution.

2.2. Pose Estimation

Using the standard pinhole camera model, we can obtain the projection $l_{i-j} (u_{i-j} \ v_{i-j})$ of the line L_{i-j} as follows.

$$\begin{aligned} u_{i-j} &= f \frac{p_{i-jx} + k_{i-j} \cdot d_{i-x}}{p_{i-jz} + k_{i-j} \cdot d_{i-z}} \\ v_{i-j} &= f \frac{p_{i-jy} + k_{i-j} \cdot d_{i-y}}{p_{i-jz} + k_{i-j} \cdot d_{i-z}} \end{aligned} \tag{8}$$

Here, f is the focal length. If k_{i-j} goes to infinity and d_{i-z} is not zero, the projection is the vanishing point and can be written as

$$\begin{aligned} u_{i-vp} &= \lim_{k_{i-j} \rightarrow \infty} f \frac{p_{i-jx} + k_{i-j} \cdot d_{i-x}}{p_{i-jz} + k_{i-j} \cdot d_{i-z}} = f \frac{d_{i-x}}{d_{i-z}} \\ v_{i-vp} &= \lim_{k_{i-j} \rightarrow \infty} f \frac{p_{i-jy} + k_{i-j} \cdot d_{i-y}}{p_{i-jz} + k_{i-j} \cdot d_{i-z}} = f \frac{d_{i-y}}{d_{i-z}} \end{aligned} \tag{9}$$

It can be seen that the vanishing point is decided only by the direction vector of the corresponding parallel lines in the camera frame.

Through feature extraction, we can obtain the expression of the line l_{i-j} . Actually, two expressions can be obtained for each set of parallel lines. Then, we can calculate the position of the vanishing point on the image plane using the two expressions. That means $(u_{i-vp} \ v_{i-vp})$ is known. Consequently, according to Equation (9), the direction vector of the corresponding 3D lines in the camera frame can be given using

$$d_i = d_{i-z} \begin{pmatrix} \frac{u_{i-vp}}{f} & \frac{v_{i-vp}}{f} & 1 \end{pmatrix} \tag{10}$$

Then, the corresponding unit direction vector in the camera frame can be written as

$$d_{i-c} = \frac{1}{\sqrt{u_{i-vp}^2 + v_{i-vp}^2 + f^2}} (u_{i-vp} \ v_{i-vp} \ f) \tag{11}$$

It can be seen that the unit direction vector of the parallel lines in the camera frame can be determined by the corresponding vanishing point. Since the vanishing point can

be calculated, the unit direction vector d_{i-c} of the parallel lines in the camera frame S_{c1} is known. In addition, as the input, the unit direction vector d_{i-w} of the parallel lines in the world frame S_{w1} is also known. According to the rigid body transformation, an Equation can be given as follows.

$$d_{i-c} = R_{w-c} \cdot d_{i-w} \tag{12}$$

Here, R_{w-c} is the rotation matrix between the world frame S_{w1} and camera frame S_{c1} , which is unknown and contains all the parameters of pose that we require for the estimate in this paper. This equation is similar to the traditional frame transformation that is written as

$$P_c = R_{w-c} \cdot P_w + t \tag{13}$$

The meaning of Equation (13) is that a 3D point P_w in the world frame S_{w1} can be transformed to P_c in the camera frame S_{c1} through the rotation matrix R_{w-c} and the translation vector t . If we let $t = 0$, the world frame and camera frame have the same origin, and then we can assume

$$\begin{aligned} P_c &= d_{i-c} \\ P_w &= d_{i-w} \end{aligned} \tag{14}$$

Now, Equations (12) and (13) are the same. Then, we can say that Equation (12) is the transformation for the 3D point when the translation vector is zero, and the coordinate of the 3D point is equal to the value of the unit direction vector. Note that this is the paper’s key point for estimating the camera pose. In detail, here we regard the transformation between the unit direction vectors as the transformation between the 3D points is a PnP problem when the translation vector is zero. To obtain the case where the translation vector is zero, we must establish a new world frame and two virtual 3D points, as shown in Figure 3.

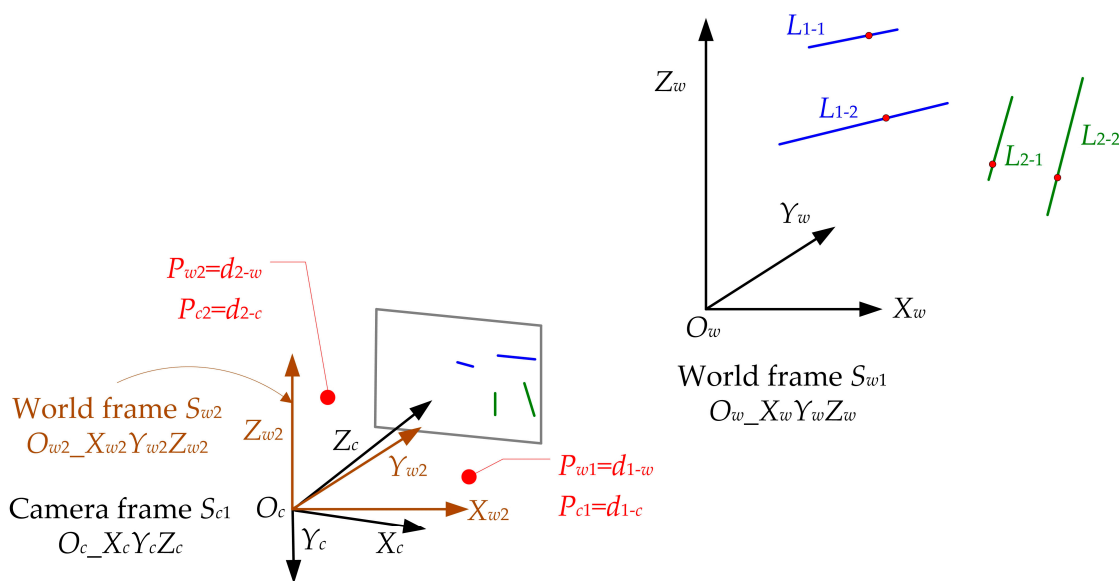


Figure 3. New world frame (brown) and two virtual 3D points (red).

A new world frame, S_{w2} ($O_{w2}X_{w2}Y_{w2}Z_{w2}$), is established in Figure 3. It is parallel to the original world frame S_{w1} , and only translation exists between the two world frames. When the origin of the world frame S_{w2} is located at the camera position O_c , we can obtain the transformation between the two world frames as follows.

$$S_{w2} = S_{w1} - O_c \tag{15}$$

In addition, according to the unit direction vectors, we established two virtual spatial points plotted in red in Figure 3. Their coordinates, both in the camera frame S_{c1} and world frame S_{w2} , are also shown in Figure 3, and then their transformation can be written as

$$P_{ci} = R_{w-c} \cdot P_{wi} \quad (16)$$

Here, R_{w-c} is both the rotation matrix between world frame S_{w2} and camera frame S_{c1} and the rotation matrix between world frame S_{w1} and camera frame S_{c1} . Next, the two virtual spatial points will be used to estimate the rotation matrix that contains all the information for the camera pose. Before that, two intermediate frames need to be established, i.e., a new world frame, S_{w3} ($O_{w3}X_{w3}Y_{w3}Z_{w3}$), and a new camera frame, S_{c2} ($O_{c2}X_{c2}Y_{c2}Z_{c2}$). The two frames coincide in space, and their origin is located at the camera position O_c , as shown in Figure 4.

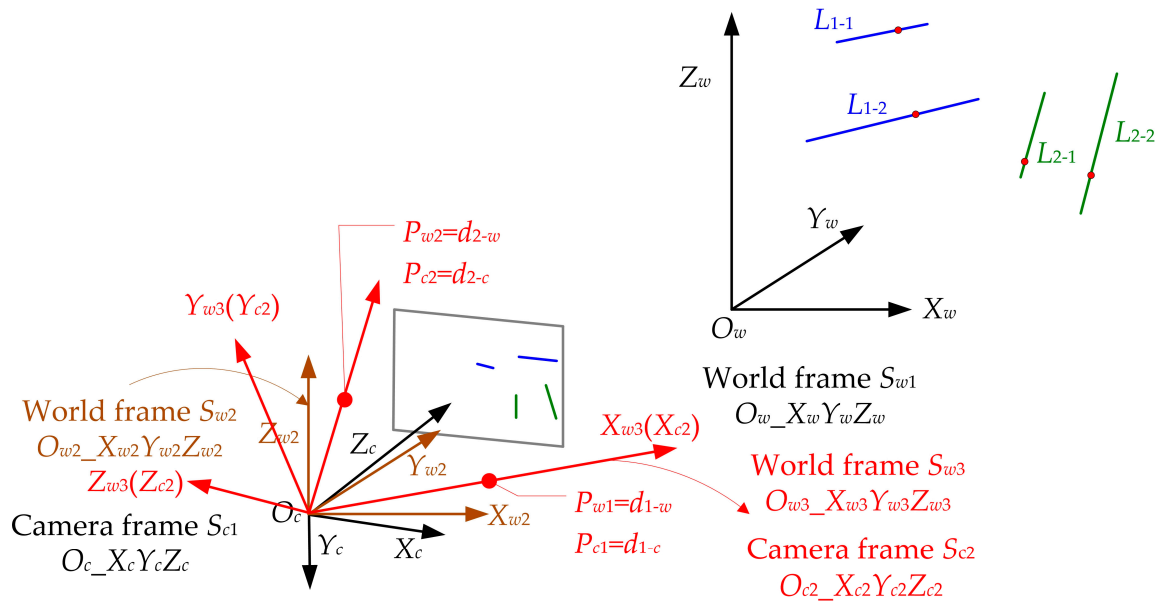


Figure 4. New camera frame S_{c2} and new world frame S_{w3} are plotted in red.

Each axis of the new camera frame S_{c2} can be calculated using

$$\begin{aligned} \vec{O_{c2}X_{c2}} &= \frac{\vec{O_cP_{c1}}}{\|\vec{O_cP_{c1}}\|} \\ \vec{O_{c2}Z_{c2}} &= \frac{\vec{O_{c2}X_{c2}} \times \vec{O_cP_{c2}}}{\|\vec{O_{c2}X_{c2}} \times \vec{O_cP_{c2}}\|} \\ \vec{O_{c2}Y_{c2}} &= \vec{O_{c2}Z_{c2}} \times \vec{O_{c2}X_{c2}} \end{aligned} \quad (17)$$

Then, the camera frame S_{c2} can be transformed into the camera frame S_{c1} using

$$\begin{aligned} S_{c2} &= T_{c-c2} \cdot S_{c1} \\ T_{c-c2} &= \begin{bmatrix} \vec{O_{c2}X_{c2}} & \vec{O_{c2}Y_{c2}} & \vec{O_{c2}Z_{c2}} \end{bmatrix}^T \end{aligned} \quad (18)$$

Each axis of the new world frame S_{w3} can be calculated using

$$\begin{aligned} \vec{O_{w3}X_{w3}} &= \frac{\vec{O_cP_{w1}}}{\|\vec{O_cP_{w1}}\|} \\ \vec{O_{w3}Z_{w3}} &= \frac{\vec{O_{w3}X_{w3}} \times \vec{O_cP_{w2}}}{\|\vec{O_{w3}X_{w3}} \times \vec{O_cP_{w2}}\|} \\ \vec{O_{w3}Y_{w3}} &= \vec{O_{w3}Z_{w3}} \times \vec{O_{w3}X_{w3}} \end{aligned} \quad (19)$$

Then, the world frame S_{w3} can be transformed into the world frame S_{w2} using

$$S_{w3} = T_{w2-w3} \cdot S_{w2}$$

$$T_{w2-w3} = \begin{bmatrix} O_{w3} \vec{X}_{w3} & O_{w3} \vec{Y}_{w3} & O_{w3} \vec{Z}_{w3} \end{bmatrix}^T \quad (20)$$

Now, we have obtained the transformations between different frames, as shown in Figure 5.

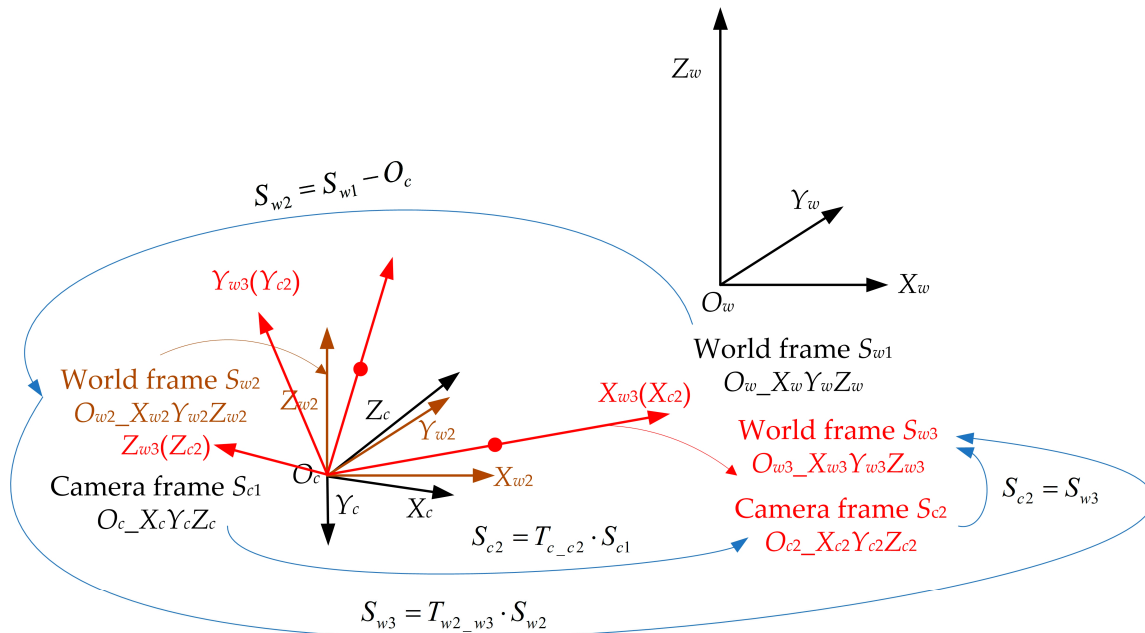


Figure 5. Transformations between different frames.

According to the known transformations between different frames, the pose estimation, that is, the transformation from world frame S_{w1} to camera frame S_{c1} , can be given using

$$S_{c1} = T_{w-c} \cdot S_{w1} + t_{w-c}$$

$$T_{w-c} = T_{c-c2}^{-1} \cdot T_{w2-w3} \quad (21)$$

$$t_{w-c} = -T_{c-c2}^{-1} \cdot T_{w2-w3} \cdot O_c$$

Now, the pose estimation is finished. Note that the solving process is similar to the method proposed in [38] but has an essential difference, which will be discussed in Section 4.

3. Experiments and Results

In this Section, first, we will thoroughly and directly test our proposed method with synthetic mass data, including numerical stability, noise sensitivity and computational speed of both pose and the focal length estimation. Simultaneously, the performance of our proposed method will be compared with that of some other existing SOTA (state-of-the-art) solvers (i.e., P3P [16], P3L [11], GPnPf (the Gauss–Newton method for the perspective-n-point and focal length) [20], RPnP (the robust O(n) solution to the perspective-n-point) [6], and DLT [23]), which involves a nonlinear algorithm, linear algorithm, point-based algorithm and line-based algorithm. In addition, some can only estimate the pose, and some can estimate both the pose and the focal length.

Second, the prior knowledge (i.e., camera position) used in our proposed method cannot be absolutely correct, which may affect the accuracy of the pose and focal length estimation, seriously or not. Hence, the robustness of our proposed method of camera position noise needs to be tested.

Last, we indirectly evaluate the performance of our proposed method with real images and compare it with the SOTA solvers to show if it can work well with real scenarios or not.

3.1. Synthetic Data

Here, synthetic mass data is generated by a virtual perspective camera with a standard pinhole camera model, whose resolution is 1280×800 , the principal point is the center of the image, and the pixel size is $14 \mu\text{m}$. In order to simplify the experiments, no distortion was added to the image, and this is reasonable in many cases where a short lens and fisheye lens are not used. Another reason is that the manufacturing and installation of the lens are both accurate, which means the distortion is small. The camera is located at $[2, 2, 2]$ in meters in the world frame, and the focal length is 50 mm.

For the P3P, GPnPf, RPnP and DLT solvers, 2D–3D point correspondences are needed. For the P3L solver, 2D–3D line correspondences are needed, and for our proposed method, 2D vanishing point–3D parallel line correspondences are needed. Hence, random 3D points, lines and parallel lines are generated in a box of $[-17, 17] \times [-11, 11] \times [50, 60]$ in meters in the camera frame. Then, the 2D correspondences are generated through the virtual perspective camera. The numbers of the 2D–3D correspondences are all three thousand for all the methods in this paper. Now, the synthetic data is generated and contains three thousand 2D–3D point correspondences, three thousand 2D–3D line correspondences, and three thousand 2D vanishing point–3D parallel line correspondences.

In this section, according to the minimal set of 2D–3D correspondences for each method, three 2D–3D point correspondences, four 2D–3D point correspondences, five 2D–3D point correspondences, six 2D–3D point correspondences, three 2D–3D line correspondences and two 2D vanishing point–3D line correspondences are randomly selected from the synthetic data for P3P, GPnPf, RPnP, DLT, and P3L and our proposed method, respectively, for each trial.

3.1.1. Robustness to Camera Position Noise

The camera position can be measured by equipment mounted on a camera, such as the IMU (inertial measurement unit) and RTK (real-time kinematic), or by other tools, such as the total station. They have high positioning accuracy, better than 3 cm [19]. In this section, we want to know how the camera position noise affects the accuracy of our proposed method because the camera position is prior knowledge, which differs from other methods. Hence, we solely analyze the robustness of the camera position noise for our proposed method.

Here, Gaussian noise, whose deviation level varies from 0 to 3 cm, is added to the camera position. For each noise level, 10,000 random trials are independently performed. Then, the mean errors of rotation, translation, reprojection and focal length are reported in Figure 6.

In Figure 6, the rotation error and focal length error are both low, which can be regarded as zero, even though error spikes exist. The reason is that the rotation and focal length estimation do not involve the camera position, which can be explained with Equations (6) and (21). In addition, as the camera position noise increases, so do the translation and reprojection errors. The reason is that the translation estimation involves the camera position, which can be explained with Equation (21). For reprojection, it is related to rotation, focal length and translation; hence, it is affected by the camera's position. When the camera position noise is 3 cm, the translation and reprojection errors both reach the maximums, which are 0.028 m and 0.26 pixels, respectively. The errors are both small and show that our proposed method has strong robustness to the camera position noise.

3.1.2. Numerical Stability

We tested our proposed method in terms of numerical stability in this section. A total of 10,000 trials were performed independently using synthetic data with no noise added. The performance of the rotation, translation and projection estimation was compared to the other five methods, and the performance of the focal length estimation was compared only to the GPnPf method because the other four methods could not estimate the focal length. The results of numerical stability are reported in Figure 7.

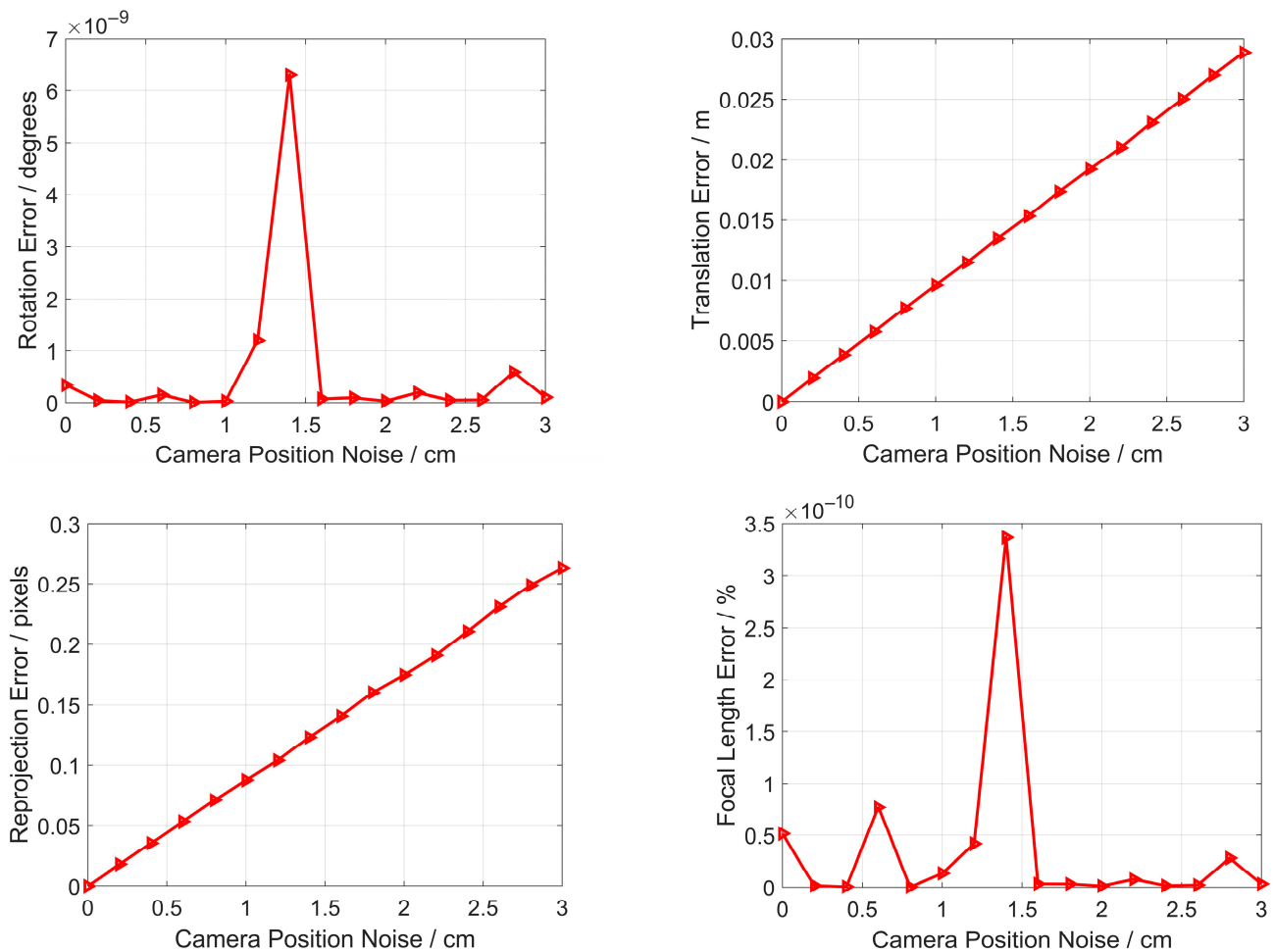


Figure 6. Robustness to camera position noise for the rotation error (Top left), translation error (Top right), reprojection error (Bottom left) and focal length error (Bottom right).

Figure 7 shows the distribution of the rotation, translation, projection and focal length error, and all six methods have good numerical stability. To be specific, the DLT method has the best performance, and our proposed method has the second in terms of rotation error; our proposed method has the best performance, and the P3L method has the second in terms of translation error; the RPnP method has the best performance, and our proposed method has the fourth in terms of reprojection error; our proposed method has the best performance, and the GPnPf method has the second in terms of focal length error. As a whole, our proposed method has the best performance in terms of numerical stability.

3.1.3. Noise Sensitivity

We tested our proposed method in terms of noise sensitivity in this section. The trials were performed independently using synthetic data with noise added. Noise may exist in the 2D feature or 3D feature. Because the 3D feature will be transformed into a 2D feature, 2D noise can reflect the 3D noise. Hence, we only added zero-mean Gaussian noise onto the 2D points and lines, and the noise deviation level varies from 0 to 1 pixel. A total of 10,000 trials were performed independently for each method, respectively, and the performance of the rotation, translation and projection estimation was compared to the other five methods. The performance of the focal length estimation was compared only to the GPnPf method because the other four methods could not estimate the focal length. The results of noise sensitivity are reported in Figure 8.

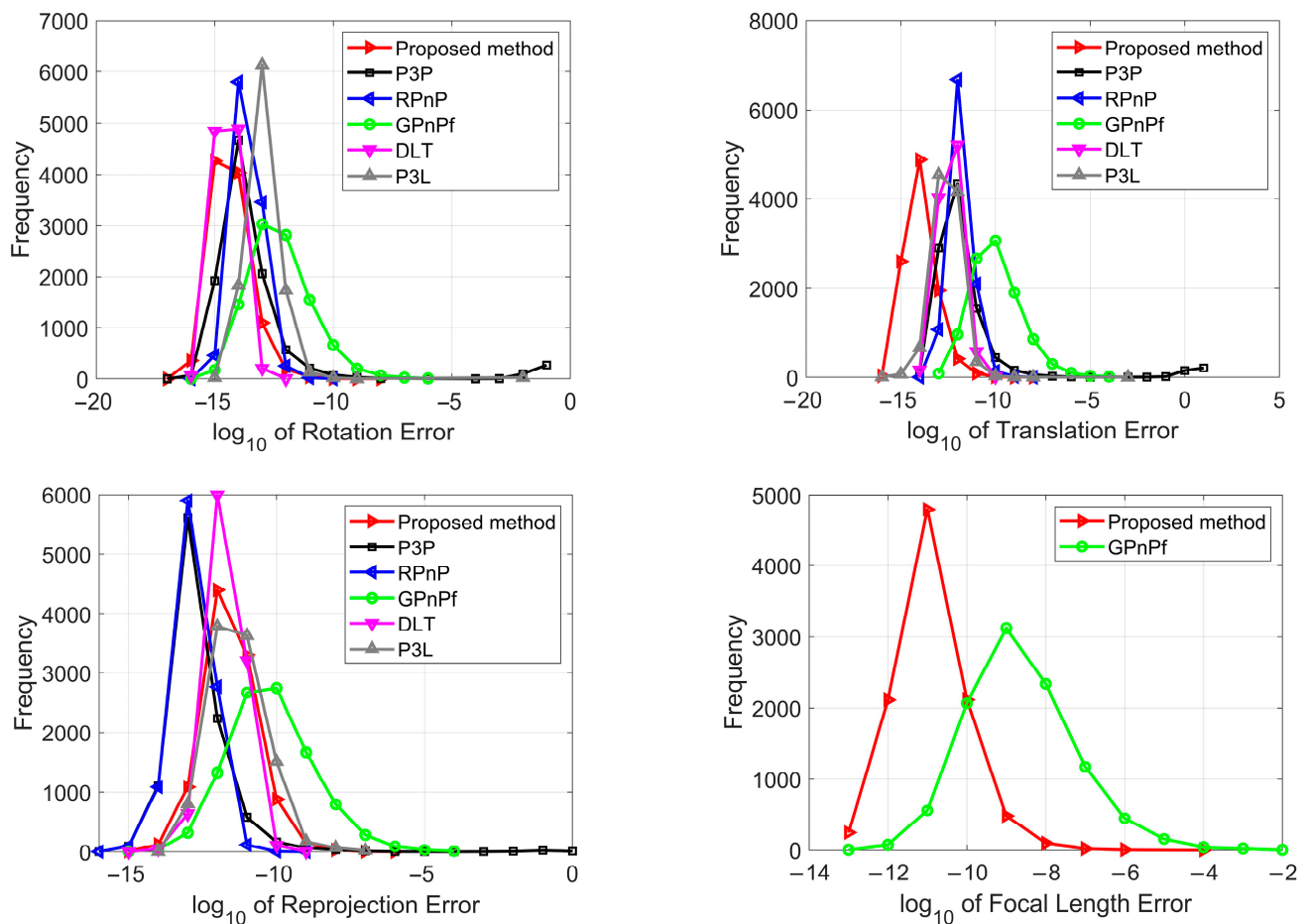


Figure 7. Numerical stability for our proposed method (red), P3P (black), RPnP (blue), GPnPf (green), DLT (purple) and P3L (gray). The (top left) is the rotation error, the (top right) is the translation error, the (bottom left) is the reprojection error, and the (bottom right) is the focal length error.

From Figure 8, it can be seen that as the noise increases, so does the rotation error, the translation error, the reprojection error and the focal length error. To be specific, the RPnP, DLT and our proposed method have similar performances, and the RPnP method performs slightly better than our proposed method in terms of rotation error; our proposed method has the best performance, and the RPnP method has the second in terms of translation error, and they both perform much better than the other four methods. The RPnP and P3P methods have similar performances, and both perform better than the other four methods in terms of reprojection error. In addition, our proposed method has the third in terms of reprojection error. Our proposed method has the best performance, and the GPnPf method has the second in terms of focal length error, and our proposed method performs much better than the GPnPf method. As a whole, our proposed method has the best or second performance in terms of noise sensitivity.

3.1.4. Computational Speed

In this section, 10,000 independent trials using synthetic data with no noise added were conducted on a 3.3 GHz two-core laptop for all six methods, respectively, to test the computational speed. Then, the mean computational times are reported in Table 1.

From Table 1, we can see that our proposed method has the best performance in terms of computational speed, and the DLT has the second. Specifically, our proposed method's computational speed is 3.2 times, 3.8 times, 14.6 times, 1.5 times and 3.0 times that of the latter five methods, respectively. This shows that our proposed method has

fast computational speed while having a good performance of numerical stability and noise sensitivity.

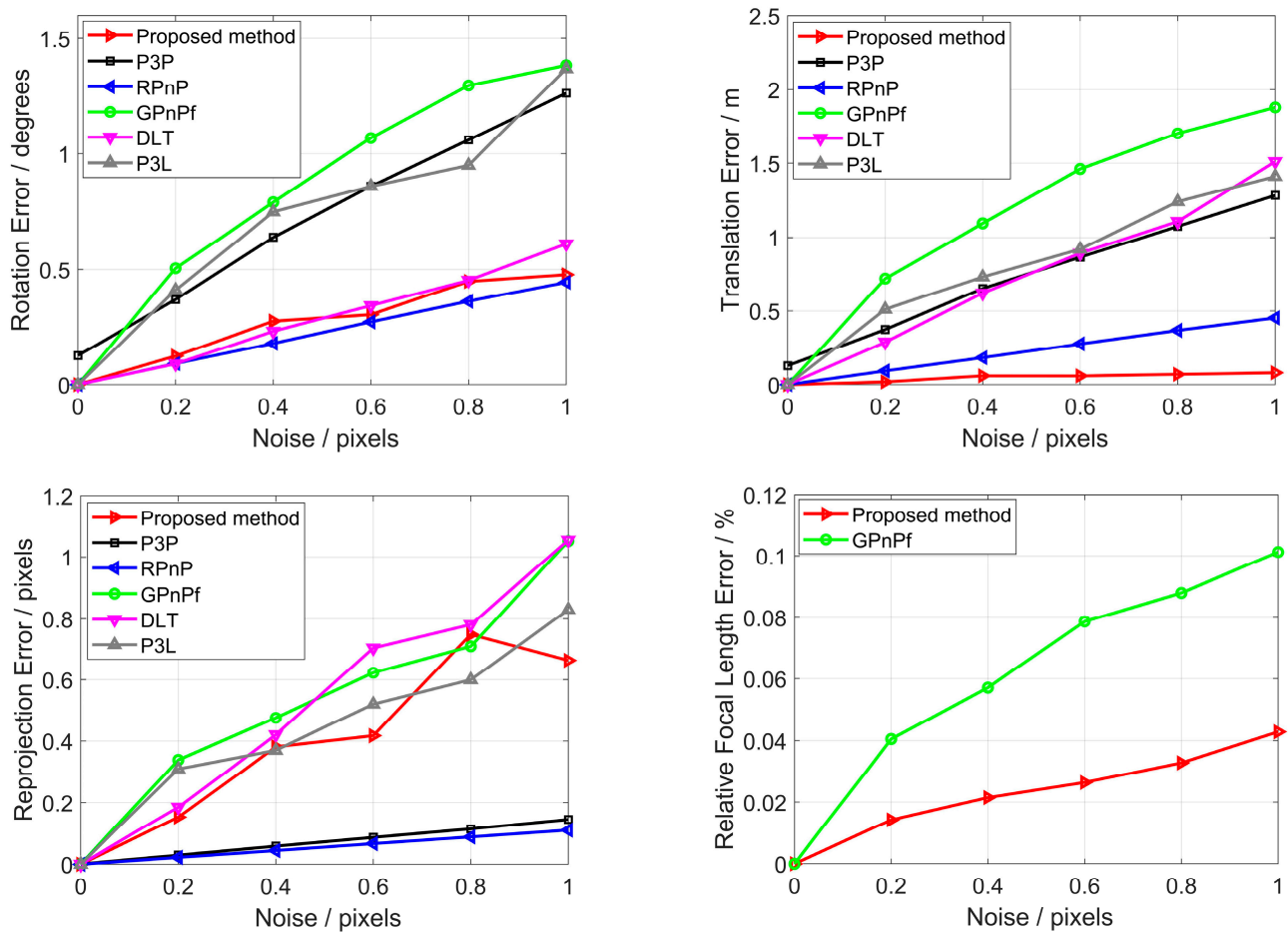


Figure 8. Noise sensitivity for our proposed method (red), P3P (black), RPnP (blue), GPnPf (green), DLT (purple) and P3L (gray). The (top left) is the rotation error, the (top right) is the translation error, the (bottom left) is the reprojection error, and the (bottom right) is the focal length error.

Table 1. Computational time.

Method	Our Proposed Method	P3P	RPnP	GP4Pf	DLT	P3L
Computational time	0.43 ms	1.37 ms	1.64 ms	6.26 ms	0.65 ms	1.31 ms

3.2. Real Images

In Section 3.1, we have shown that our proposed method can work well with synthetic data directly. To fully test our proposed method, we will now use real images to show whether it works well with real scenarios. The cameras were placed in real scenarios, but the ground truths of their poses are not known. This problem suggests that we cannot directly test our proposed method. Here, an indirect method was established to test it.

First, many lines and points, whose positions are known as ground truth, were placed in the FOV, and then we chose some of them to estimate the pose and focal length for our proposed method and other SOTA methods. After estimating, stereo vision [43] was used to measure the 3D positions of the left points as the measured values. The accuracy of the measured value is affected by the camera's pose and focal length. The pose and focal length were estimated by our proposed method and other SOTA methods. Hence, the measurement accuracy between the measured value and ground truth can reflect the

accuracy of our proposed method. Next, we set up the real scenarios and captured real images from two different views by the cameras [38], shown in Figure 9.

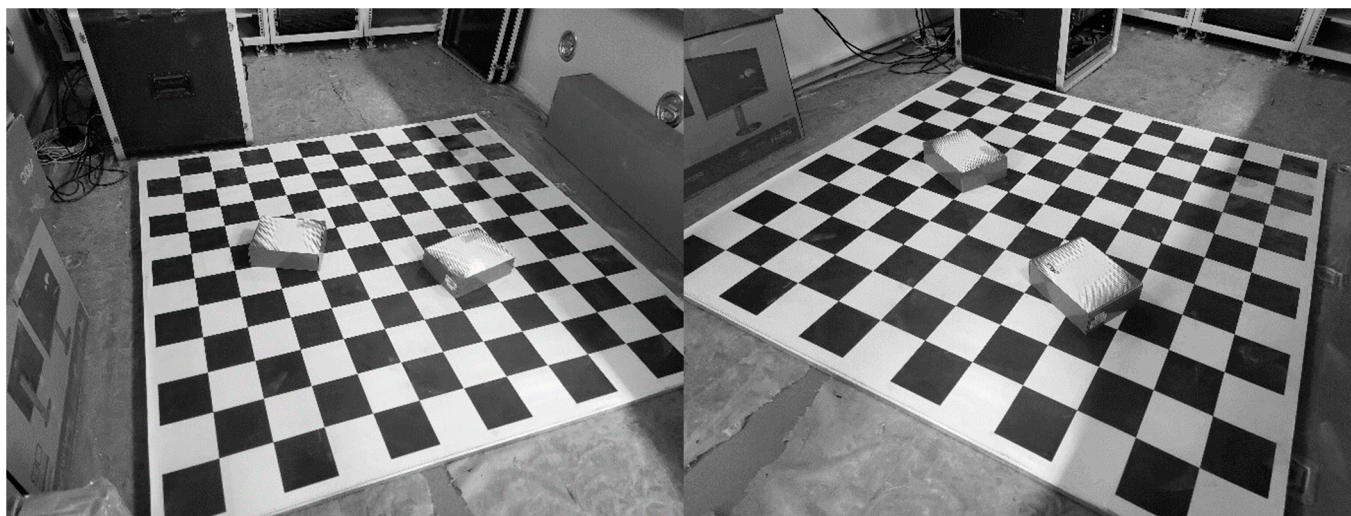


Figure 9. Real images from two different views.

The checkerboard was placed in the FOV, and the size was known. Hence, there are many sets of parallel lines, and their unit direction vectors are known. We chose two sets for our proposed method to estimate the focal length and pose, as shown in Figure 10.

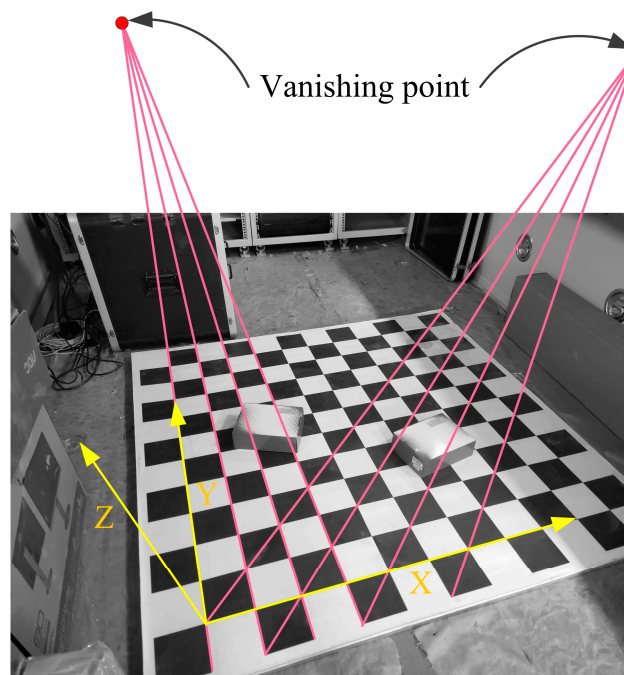


Figure 10. Two vanishing points were extracted for the focal length and pose estimation.

The world frame (yellow) was established, as shown in Figure 10, and two vanishing points were obtained from the two sets of parallel lines (red). In addition, for the P3P, GPnPf, RPnP, DLT and P3L solvers, three points, four points, five points, six points and three lines from the checkerboards were randomly chosen to estimate the focal length and pose. The camera positions were measured by a total station for our proposed method. After the focal length and pose estimations, the stereo vision was used to measure the positions of the left points on the checkerboards as measured values. Then, we obtained the

mean relative position errors between the measured values and ground truths, as shown in Table 2. Moreover, the reprojections of the left points can then be obtained, and the mean reprojection errors between the reprojection and projection are also reported in Table 2.

Table 2. Mean relative position errors and mean reprojection errors.

Method	Proposed Method	P3P	RPnP	GPnPf	DLT	P3L
Mean relative error %	0.45	0.54	1.81	1.17	0.59	0.72
Mean reprojection/pixel	0.61	0.56	0.49	0.79	0.72	0.67

Our proposed method and P3P have the best performance in terms of the mean relative position error; our proposed method has the third performance, and RPnP has the first in terms of the mean reprojection error. As a whole, our proposed method performs best in real images.

In addition, we obtained the computational time of all the methods and our proposed method has the best performance. Specifically, the computational speed of our proposed method is 2.5 times, 3.4 times, 14.2 times, 1.6 times and 3.1 times that of the latter five methods, respectively. This is basically consistent with the results in the synthetic data.

For real images, there are many factors here that affect our proposed method, such as noise and error in the camera's position. In fact, we took these factors into account when setting up our real scenario in this section. When extracting the feature points, we used the sub-pixel extraction algorithm, which will introduce noise. This is also related to the imaging quality, so the noise cannot be quantified. In addition, in the experiment, we used the total station to obtain the camera position because the total station itself has an error; hence, the error of the camera position is also introduced in the real scenario. It is clear that we should consider the influence of these factors when analyzing the performance of our proposed method for the real scenario. Under the influence of these factors, our method still shows good performance. In addition, due to the real scenario limitations, it is difficult to set up many real scenarios. Therefore, we adopted synthetic data to simulate different scenarios in Section 3.1. Using a combination of a small number of real scenarios and a large number of synthetic scenarios, and in the case of introducing various types of errors, shows our proposed method has good performance.

4. Discussion

This paper uses two vanishing points and a camera's position to estimate the focal length and pose simultaneously. To our best knowledge, this is the first paper to perform this job using vanishing points and a camera's position. Using the camera position as the prior knowledge can simplify the estimation problem and improve accuracy and efficiency. Unlike other existing methods, our proposed method does not involve nonlinear computation and multi-solution phenomenon and needs only two vanishing points. In computer vision, our proposed method can estimate the camera pose more quickly in the case of multiple vanishing points. It is another idea to be used to estimate the pose, which is complementary to other calibration methods. The differences and advantages of the proposed method and future work will be discussed as follows.

4.1. Differences and Advantages

The first difference is that our proposed method uses the camera position as the prior knowledge, and it can simplify the problem. In fact, many existing methods also use some prior knowledge and also simplify the problem. However, the costs of using prior knowledge for different methods are different. Additionally, the corresponding benefits are different. Some methods require expensive equipment with large sizes and precision mechanical structures to obtain prior knowledge and might not achieve a very good effect. However, the camera's position can be given by RTK, which is cheap and has a small size. In addition, it has strong robustness (Section 3.1.1) to the camera's position and good

performance in terms of numerical stability, noise sensitivity and computational speed (Sections 3.1.2–3.1.4). This means we can obtain good benefits at a low cost. The advantage can also be seen indirectly in Section 3.2. Last, the rotation and focal length estimation do not involve the camera position; hence, they have no error when the camera noise exists (Section 3.1.1).

The second difference is that our proposed method does not involve nonlinear iterations. Some existing methods, e.g., P3P, RPnP and GPnPf, need to solve the nonlinear equation, and in order to avoid the optimal local solution, iteration is needed. Although nonlinear iteration could improve the accuracy, the computational speed is decreased, as shown in Section 3.1.4. This is the main reason why our proposed method and DLT have the best performance in terms of computational speed. In addition, improving the accuracy does not mean we can always obtain the optimal global solution, and this leads to our proposed method perhaps having higher accuracy, as shown in Section 3.1.3.

The third difference is that our proposed method has no multi-solution phenomenon. When we estimate the focal length, a quadratic equation with one unknown must be solved. When we estimate the pose, the computational process mainly involves multiplication and matrix operations. Hence, there is no multi-solution phenomenon. Because one more constraint is needed to disambiguate the multi-solution phenomenon, the computational speed will decrease, and this is another reason why our proposed method has the best performance in terms of computational speed.

Last, the calculating process is similar to another method that we proposed in [38]; however, they have essential differences. The method in [38] needs to establish two planes in the world frame and camera frame and obtain their normal unit vectors. However, this paper directly uses the unit direction vectors of lines and does not establish the planes. Additionally, when we estimate the focal length, this paper uses the angle between two lines, not the two planes in [38]. This paper only requires the unit direction vectors of the 3D lines but does not need the positions in the space. The method in [38] needs both. It can be seen (although the forms of the equations and calculating process are similar) that the meanings of the two methods are totally different.

The main disadvantage is that our proposed method does not perform best in terms of reprojection error. The reason is that some other methods, such as P3P and RPnP, use iteration to refine the solution, and the corresponding cost function is to make the reprojection error minimal. Our proposed method has no refining process, which leads to the main disadvantage.

Briefly, our proposed method has the following advantages. (1) Only two vanishing points are needed; (2) it has no multi-solution phenomenon; (3) it has strong robustness to camera noise; (4) as a whole, it performs well in terms of numerical stability and noise sensitivity; (5) the computational speed is fast. The main disadvantage is that it does not perform best in terms of reprojection error.

4.2. Future Work

As described in Section 4.1, our proposed method has a main disadvantage for reprojection error. Hence, the main work in the future is to establish the cost function to minimize the reprojection error and simultaneously refine the solution. Another work will use other sensors to obtain additional prior knowledge, such as the IMUs, which can give two orientations of the camera. It also can simplify the problem. Additionally, it may be possible that the camera's position and partial orientation are both used as prior knowledge to improve accuracy or estimate more intrinsic parameters.

5. Conclusions

This paper proposed a new method to estimate the focal length and pose based on two vanishing points and a camera's position. The key point is to convert the transformation between the unit direction vectors to the transformation between the 3D points without

translation. The experimental results show that, as a whole, our proposed method performs better than some existing state-of-the-art methods.

Author Contributions: Conceptualization, K.G. and R.C.; methodology, K.G.; software, K.G. and Y.T.; validation, B.J.; formal analysis, K.G. and X.D.; investigation, X.L.; resources, K.G.; data curation, Y.T.; writing—original draft preparation, R.C.; writing—review and editing, K.G.; visualization, B.J. and K.G.; supervision, X.D.; project administration, X.L.; funding acquisition, R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Elqursh, A.; Elgammal, A. Line-based relative pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 3049–3056.
2. Hee Lee, G.; Pollefeys, M.; Fraundorfer, F. Relative pose estimation for a multi-camera system with known vertical direction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 540–547.
3. Nistér, D. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 756–770. [[CrossRef](#)] [[PubMed](#)]
4. Martyushev, E.; Li, B. Efficient relative pose estimation for cameras and generalized cameras in case of known relative rotation angle. *J. Math. Imaging Vis.* **2020**, *62*, 1076–1086. [[CrossRef](#)]
5. Guan, B.; Zhao, J.; Li, Z.; Sun, F.; Fraundorfer, F. Minimal solutions for relative pose with a single affine correspondence. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1929–1938.
6. Li, S.; Xu, C.; Xie, M. A robust O(n) solution to the perspective-n-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1444–1450. [[CrossRef](#)] [[PubMed](#)]
7. Wang, P.; Xu, G.; Cheng, Y.; Yu, Q. A simple, robust and fast method for the perspective-n-point problem. *Pattern Recognit. Lett.* **2018**, *108*, 31–37. [[CrossRef](#)]
8. Zheng, Y.; Kuang, Y.; Sugimoto, S.; Astrom, K.; Okutomi, M. Revisiting the pnp problem: A fast, general and optimal solution. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2344–2351.
9. Wu, Y.; Hu, Z. PnP problem revisited. *J. Math. Imaging Vis.* **2006**, *24*, 131–141. [[CrossRef](#)]
10. Zhou, L.; Koppel, D.; Kaess, M. A complete, accurate and efficient solution for the perspective-n-line problem. *IEEE Robot. Autom. Lett.* **2020**, *6*, 699–706. [[CrossRef](#)]
11. Xu, C.; Zhang, L.; Cheng, L.; Koch, R. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1209–1222. [[CrossRef](#)]
12. Příbyl, B.; Zemčík, P.; Čadík, M. Absolute pose estimation from line correspondences using direct linear transformation. *Comput. Vis. Image Underst.* **2017**, *161*, 130–144. [[CrossRef](#)]
13. Lee, G.H. A minimal solution for non-perspective pose estimation from line correspondences. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 170–185.
14. Lee, S.C.; Jung, S.K.; Nevatia, R. Automatic pose estimation of complex 3D building models. In Proceedings of the IEEE Workshop on Applications of Computer Vision, Orlando, FL, USA, 3–4 December 2002; pp. 148–152.
15. Guillou, E.; Meneveaux, D.; Maisel, E.; Bouatouch, K. Using vanishing points for camera calibration and coarse 3D reconstruction from a single image. *Vis. Comput.* **2000**, *16*, 396–410. [[CrossRef](#)]
16. Ke, T.; Roumeliotis, S.I. An efficient algebraic solution to the perspective-three-point problem. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7225–7233.
17. Kneip, L.; Scaramuzza, D.; Siegwart, R. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 2969–2976.
18. Gao, X.S.; Hou, X.R.; Tang, J.; Cheng, H.F. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 930–943.
19. Guo, K.; Ye, H.; Gu, J.; Chen, H. A novel method for intrinsic and extrinsic parameters estimation by solving perspective-three-point problem with known camera position. *Appl. Sci.* **2021**, *11*, 6014. [[CrossRef](#)]

20. Zheng, Y.; Sugimoto, S.; Sato, I.; Okutomi, M. A general and simple method for camera pose and focal length determination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 430–437.
21. Bujnak, M.; Kukulova, Z.; Pajdla, T. A general solution to the P4P problem for camera with unknown focal length. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 24–26 June 2008; pp. 1–8.
22. Kukulova, Z.; Bujnak, M.; Pajdla, T. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2816–2823.
23. Quan, L.; Lan, Z. Linear n-point camera pose determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 774–780. [[CrossRef](#)]
24. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
25. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2003.
26. Kukulova, Z.; Bujnak, M.; Pajdla, T. Closed-form solutions to minimal absolute pose problems with known vertical direction. In Proceedings of the Asian Conference on Computer Vision, Queenstown, New Zealand, 8–12 November 2010; pp. 216–229.
27. Svärm, L.; Enqvist, O.; Kahl, F.; Oskarsson, M. City-scale localization for cameras with known vertical direction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1455–1461. [[CrossRef](#)] [[PubMed](#)]
28. Albl, C.; Kukulova, Z.; Pajdla, T. Rolling shutter absolute pose problem with known vertical direction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3355–3363.
29. D’Alfonso, L.; Garone, E.; Muraca, P.; Pugliese, P. P3P and P2P Problems with known camera and object vertical directions. In Proceedings of the Mediterranean Conference on Control and Automation, Crete, Greece, 25–28 June 2013; pp. 444–451.
30. Guo, K.; Ye, H.; Chen, H.; Gao, X. A New Method for Absolute Pose Estimation with Unknown Focal Length and Radial Distortion. *Sensors* **2022**, *22*, 1841. [[CrossRef](#)]
31. Guo, K.; Ye, H.; Zhao, Z.; Gu, J. An efficient closed form solution to the absolute orientation problem for camera with unknown focal length. *Sensors* **2021**, *21*, 6480. [[CrossRef](#)]
32. Guo, K.; Ye, H.; Gao, X.; Chen, H. An Accurate and Robust Method for Absolute Pose Estimation with UAV Using RANSAC. *Sensors* **2022**, *22*, 5925. [[CrossRef](#)]
33. Wang, P.; Xu, G.; Cheng, Y. A novel algebraic solution to the perspective-three-line pose problem. *Comput. Vis. Image Underst.* **2020**, *191*, 102711. [[CrossRef](#)]
34. Zhang, L.; Xu, C.; Lee, K.M.; Koch, R. Robust and efficient pose estimation from line correspondences. In Proceedings of the Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012; pp. 217–230.
35. Ansar, A.; Daniilidis, K. Linear pose estimation from points or lines. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 578–589. [[CrossRef](#)]
36. Lecrosnier, L.; Boutteau, R.; Vasseur, P.; Savatier, X.; Fraundorfer, F. Vision based vehicle relocalization in 3D line-feature map using Perspective-n-Line with a known vertical direction. In Proceedings of the IEEE Intelligent Transportation Systems Conference, Auckland, New Zealand, 27–30 October 2019; pp. 1263–1269.
37. Horanyi, N.; Kato, Z. Multiview absolute pose using 3D-2D perspective line correspondences and vertical direction. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 2472–2480.
38. Guo, K.; Zhang, Z.; Zhang, Z.; Tian, Y.; Chen, H. Fast and Accurate Pose Estimation with Unknown Focal Length Using Line Correspondences. *Sensors* **2022**, *22*, 8253. [[CrossRef](#)]
39. Caprile, B.; Torre, V. Using vanishing points for camera calibration. *Int. J. Comput. Vis.* **1990**, *4*, 127–139. [[CrossRef](#)]
40. He, B.W.; Li, Y.F. Camera calibration from vanishing points in a vision system. *Opt. Laser Technol.* **2008**, *40*, 555–561. [[CrossRef](#)]
41. Grammatikopoulos, L.; Karras, G.; Petsa, E. Camera calibration combining images with two vanishing points. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *35*, 99–104.
42. Guo, K.; Ye, H.; Gu, J.; Tian, Y. A Fast and Simple Method for Absolute Orientation Estimation Using a Single Vanishing Point. *Appl. Sci.* **2022**, *12*, 8295. [[CrossRef](#)]
43. Tippetts, B.; Lee, D.J.; Lillywhite, K.; Archibald, J. Review of stereo vision algorithms and their suitability for resource-limited systems. *J. Real-Time Image Process.* **2016**, *11*, 5–25. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.