

Article

Pose Estimation of Omnidirectional Camera with Improved EPnP Algorithm

Xuanrui Gong, Yaowen Lv *, Xiping Xu, Yuxuan Wang and Mengdi Li

College of Optoelectronic Engineering, Changchun University of Science and Technology, Changchun 130022, China; 2019200040@mails.cust.edu.cn (X.G.); xxp@cust.edu.cn (X.X.); 2019100213@mails.cust.edu.cn (Y.W.); 2017100212@mails.cust.edu.cn (M.L.)

* Correspondence: lyw@cust.edu.cn; Tel.: +86-1504-4100-195

Abstract: The omnidirectional camera, having the advantage of broadening the field of view, realizes 360° imaging in the horizontal direction. Due to light reflection from the mirror surface, the collinearity relation is altered and the imaged scene has severe nonlinear distortions. This makes it more difficult to estimate the pose of the omnidirectional camera. To solve this problem, we derive the mapping from omnidirectional camera to traditional camera and propose an omnidirectional camera linear imaging model. Based on the linear imaging model, we improve the EPnP algorithm to calculate the omnidirectional camera pose. To validate the proposed solution, we conducted simulations and physical experiments. Results show that the algorithm has a good performance in resisting noise.

Keywords: omnidirectional camera; pose estimation; Perspective-n-Point



Citation: Gong, X.; Lv, Y.; Xu, X.; Wang, Y.; Li, M. Pose Estimation of Omnidirectional Camera with Improved EPnP Algorithm. *Sensors* **2021**, *21*, 4008. <https://doi.org/10.3390/s21124008>

Academic Editor: Jose Manuel Menéndez García

Received: 7 March 2021
Accepted: 17 May 2021
Published: 10 June 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/>).

1. Introduction

The Perspective-n-Point (PnP) problem is a classic problem in computer vision. The aim is to calculate the orientation of a camera given its intrinsic parameter and a set of correspondences between 3D points and their 2D points. It is widely used in computer vision. In 2008, Xu [1] used an auxiliary point to extend the linear method for the special case of four coplanar points and find the coarse solutions for the general P3P problem. In 2012, Li [2] proposed a solution for the PnP problem that can retrieve the optimum by solving a seventh-order polynomial. In 2018, Wang [3] transferred the problem to solve a seventh-order and a fourth-order univariate polynomial. In 2020, Zhou [4] constructed the PnP problem for an uncalibrated camera as a 20th-order polynomial system. In 2021, Meng [5] mitigated scale bias by multiplying an independent inverse average depth variable onto the object space error to improve the accuracy of pose estimation.

Recently, omnidirectional vision system has become a hot topic for researchers in some fields such as robot driving [6,7], augmented reality [8], and video surveillance [9–11]. An omnidirectional vision system provides a 360-degree panorama in the horizontal direction and is composed of a common CCD and mirror. It can help reduce the number of cameras needed and overall costs. For omnidirectional camera pose estimation, the control point coordinates around the camera improve the accuracy of camera pose estimation. A large field of view can effectively reduce the loss of tracking caused by matching. So, the problem of pose estimation based on an omnidirectional vision system is important. In 2001, Daniel G. Aliaga [12], an earlier researcher in this research direction, proposed a pose estimation algorithm based on coded points. He set up a complete stand-alone system, but the method is only applicable for parabolic mirror omnidirectional camera. In the same year, Paulino [13] proposed a method to estimate the pose of a center omnidirectional camera with an arbitrary mirror surface. Gebken [14] presented a novel perspective pose estimation for omnidirectional vision that involves a parabolic central catadioptric sensor by using small data sets, combining geometry and stochastics such that they obtain pose

from only three image points. Goncalves [15] proved that any reflection point belongs to an analytical quadric that intersects the mirror quadric itself and presented a linear method to estimate the pose for the noncentral omnidirectional system. Ilizirov [16] proposed deriving the metric between internal reflections; a closed form similar to the principle of collinearity was obtained and then extended to a linear model. Miraldo [17] derived parametric equations to estimate pose for vanishing points and vanishing curves by using the calibration parameters and mirror shape coefficients. Due to the nonlinear imaging of omnidirectional cameras, a high-order constraint or more complicated formula is needed to establish the imaging model or camera pose estimate. In this paper, we establish a relationship between the omnidirectional image and the traditional camera image to simplify the imaging system.

In 2009, Lepetit [18] proposed the EPnP algorithm, which is widely used to estimate the pose of the conventional transmission camera. The EPnP algorithm speeds up the calculation by reducing $O(3)$ to $O(1)$ and introduces four reference control points to reduce the influence of a single control point error. Since the EPnP algorithm was proposed, it has received widespread attention, and some researchers have also proposed algorithms derived from the EPnP algorithm [19–21]. Based on EPnP algorithm, Penatesanchez [19] proposed a pose estimation algorithm with unknown camera focal length. In the same year, Deng [20] applied the EPnP algorithm to mosaic images. In 2018, Chen Peng [21] improved the EPnP algorithm by the Gauss-Newton method to optimize the coordinates of virtual control points in the camera coordinate system.

However, in the EPnP algorithm, linear equations are established based on imaging systems to solve the camera pose. This paper analyzes the imaging model of the omnidirectional camera and presents the linear virtual imaging plane. We derive the linear imaging equation of the omnidirectional camera and put forward a method that estimates omnidirectional camera pose by the EPnP algorithm. First of all, the control points of an omnidirectional image are projected into a linear virtual image plane. Then, the control point location in the virtual camera coordinate system, solved by the EPnP algorithm, is converted into the location of the omnidirectional camera coordinate system. Finally, the omnidirectional camera pose is solved by an absolute orientation method. Section 2 describes the spherical model of an omnidirectional camera. Section 3 describes the specific algorithm. Sections 4 and 5 respectively describe simulations and real image experiments and verify the noise immunity and reliability of this method.

2. Omnidirectional Camera Spherical Model

The omnidirectional vision system is composed of a camera and a mirror. Mei presented an omnidirectional camera spherical model and corresponding calibration toolbox [22]. They proved the equivalence of the reflection process and spherical mapping.

First, the spatial point \mathbf{X}^c with coordinate $(\mathbf{X}^c, \mathbf{Y}^c, \mathbf{Z}^c)$ is projected onto the unit sphere to obtain point \mathbf{X}^m .

Then, in the new reference frame centered with $(0, 0, -\zeta)$ as the origin, \mathbf{X}^m is changed to \mathbf{X}^s .

Next, the point \mathbf{X}^s is projected onto a normalized plane to obtain the point \mathbf{u}^s .

The last projection is related to a generalized camera projection matrix \mathbf{K} . The point \mathbf{u}^s on the normalized plane is projected onto the image plane by matrix \mathbf{K} .

As shown in Figure 1, the relationship between an arbitrary 3D space point \mathbf{X}^c and corresponding image plane point \mathbf{u} can be written as:

$$\lambda \mathbf{u}_i = \mathbf{K} \cdot h \left(\frac{\mathbf{X}_i^c}{\|\mathbf{X}_i^c\|} \right) \quad (1)$$

$$\text{with } \mathbf{K} = \begin{bmatrix} f_1 \eta & f_1 \eta \alpha & u_0 \\ 0 & f_2 \eta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, h(\mathbf{X}_i^m) = \begin{bmatrix} \frac{X_i^m}{Z_i^m + \zeta} \\ \frac{Y_i^m}{Z_i^m + \zeta} \\ 1 \end{bmatrix}.$$

where $\|X_i^c\|$ is norm of X_i^c . The function h represents an expression of a spatial point to a normalized plane. ζ and η are related to the mirror of the omnidirectional camera. Table 1 shows the values of ζ and η in different mirrors. f_1 and f_2 are the focal lengths of the camera in the X and Y directions, respectively. α is the correlation coefficient between the X and Y directions and is usually equal to zero. A generalized camera projection matrix indicates we are no longer considering the sensor as a separate camera and mirror but as a global device. f and η cannot be estimated independently. We will note $f_x = f_1\eta$ and $f_y = f_2\eta$.

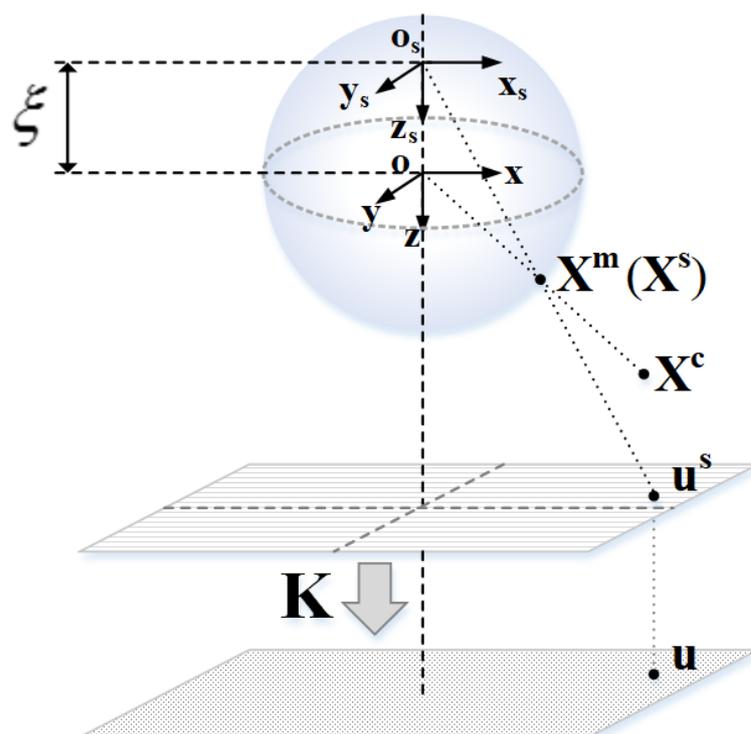


Figure 1. Unit spherical model.

Table 1. Unified model parameters.

Parameter	ζ	η
Parabola	1	$-2p$
Hyperbola	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{-2p}{\sqrt{d^2+4p^2}}$
Ellipse	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{2p}{\sqrt{d^2+4p^2}}$
Planar	0	-1

The camera calibration process can obtain the projection matrix \mathbf{K} and the coefficients ζ .

3. Pose Estimation Algorithm

The traditional camera pinhole model, camera optical center, image point, and space control point meet the collinear condition. According to the characteristic, we introduce virtual imaging that is the process of projecting virtual space point coordinates X^{vir} onto the virtual plane u^{vir} . The logic of the approach is shown in Algorithm 1. We respectively calculate the coordinate transformation from image point to virtual image point and from space point to virtual space point.

Algorithm 1. Logic of omnidirectional camera pose estimation algorithms.

1. **Data acquisition.** Extract corner point \mathbf{u} and obtain its coordinate in the world coordinate system \mathbf{X}^w
2. **Determine camera.** Determine the number and location of virtual cameras
3. **Coordinate transformation.** Convert corner coordinate to virtual image plane coordinate.
4. **Calculate \mathbf{X}^{vir} .** Use modified EPNP algorithm to calculate virtual camera coordinate \mathbf{X}^{vir} .
5. **Coordinate transformation.** Convert virtual camera coordinate to camera coordinate.
6. **Calculate pose.** Calculate the camera pose according to \mathbf{X}^c and \mathbf{X}^w

As shown in Figure 2, two virtual cameras are established to substitute the omnidirectional camera. The optical center of the virtual camera is coinciding with the optical center of the omnidirectional camera, in which optical axes are aligned with the positive and the negative directions of the Z-axis. In a virtual camera, the conversion from the image coordinate system to the pixel coordinate system is not important, so we simplify this part. The virtual image plane is actually the normalized plane in the perforation imaging model, that is, the $Z = 1$ or $Z = -1$ plane.

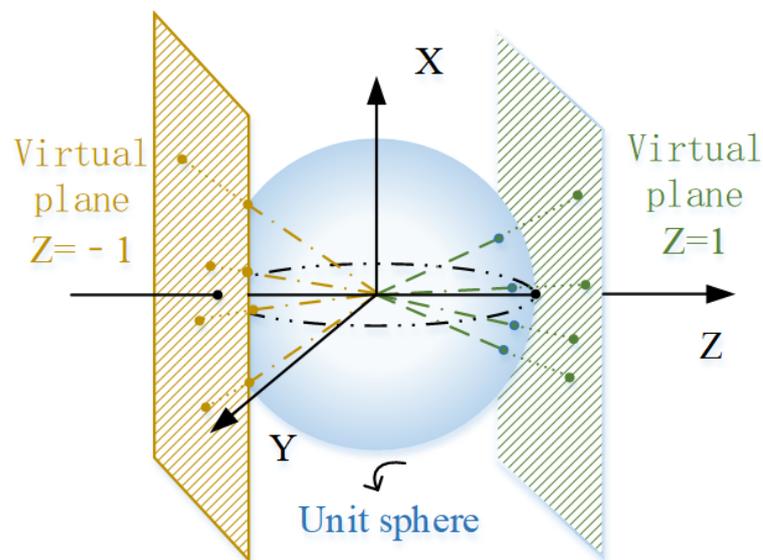


Figure 2. Virtual camera coordinate system.

After determining the number of virtual cameras and the position of the virtual image plane, the next step is to calculate the coordinates of the virtual points. First, the image point \mathbf{u} is projected onto the unit sphere. Through algebraic operations, we can obtain an equation from the image point to the corresponding unit sphere.

$$\mathbf{X}^m = h^{-1}(\mathbf{K}^{-1}\mathbf{u}) \quad (2)$$

$$h^{-1}(\mathbf{u}^s) = \begin{bmatrix} \frac{\zeta + \sqrt{1 + (1 + \zeta^2)(u_i^{s2} + v_i^{s2})}}{u_i^{s2} + v_i^{s2} + 1} \cdot u_i^s \\ \frac{\zeta + \sqrt{1 + (1 + \zeta^2)(u_i^{s2} + v_i^{s2})}}{u_i^{s2} + v_i^{s2} + 1} \cdot v_i^s \\ \frac{\zeta + \sqrt{1 + (1 + \zeta^2)(u_i^{s2} + v_i^{s2})}}{u_i^{s2} + v_i^{s2} + 1} - \zeta \end{bmatrix} \quad (3)$$

where (u^s, v^s) are the coordinates of $\mathbf{u}^{s'}$. h^{-1} is essentially the inverse of h in Equation (1). The coordinate of \mathbf{X}^m satisfies the equation $X^{m2} + Y^{m2} + Z^{m2} = 1$.

Then, we normalize the Z^m coordinate to obtain the virtual image plane coordinates.

$$\begin{bmatrix} u_i^{vir} \\ v_i^{vir} \\ 1 \end{bmatrix} = \frac{1}{Z_i^m} \begin{bmatrix} X_i^m \\ Y_i^m \\ Z_i^m \end{bmatrix} \text{ or } \begin{bmatrix} u_i^{vir} \\ v_i^{vir} \\ -1 \end{bmatrix} = \frac{1}{-Z_i^m} \begin{bmatrix} X_i^m \\ Y_i^m \\ Z_i^m \end{bmatrix} \quad (4)$$

While the virtual image Z is equal to 1, the virtual camera coordinate system will coincide with the omnidirectional camera system. If the virtual image Z is equal to -1 , there is a corresponding relationship between the virtual camera coordinate system and the omnidirectional camera coordinate system: $Z^{vir} = -Z^c$, $Y^{vir} = -X^c$, and $X^{vir} = -Y^c$. The virtual camera imaging equation can be written as:

$$\lambda^{vir} \begin{bmatrix} u_i^{vir} \\ v_i^{vir} \\ 1 \end{bmatrix} = \mathbf{K}^{vir} \mathbf{X}_i^{vir} = \mathbf{I} \mathbf{X}_i^{vir} \quad (5)$$

where the λ^{vir} is scalar depth parameters. \mathbf{K}^{vir} is the internal calibration matrix of the virtual camera. For the distance from the optical center to the image plane, $f^{vir} = 1$, and for the principal point of the virtual camera, $(u_0, v_0) = (0, 0)$. So, the virtual camera internal calibration matrix \mathbf{K}^{vir} is equivalent to \mathbf{I} .

In the next step, the virtual image points and imaging equations are brought into the EPNP algorithm to calculate the point coordinates in the virtual camera coordinate system. Suppose the control points in the world coordinate system are $\{\mathbf{X}_1^w, \mathbf{X}_2^w, \mathbf{X}_3^w, \mathbf{X}_4^w, \dots, \mathbf{X}_n^w\}$. \mathbf{X}_i^w ($i = 1 \dots n$) can be written by four reference points ($\mathbf{C}_1^w, \mathbf{C}_2^w, \mathbf{C}_3^w, \mathbf{C}_4^w$) in the world coordinate system.

$$\mathbf{X}_i^w = \sum_{j=1}^4 \alpha_{ij} \mathbf{C}_j^w \quad (6)$$

Because of the nature of Euclidean space, the same relationship holds in the virtual camera coordinate system. We define the reference points in the virtual coordinate system as $\mathbf{C}_1^{vir}, \mathbf{C}_2^{vir}, \mathbf{C}_3^{vir}, \mathbf{C}_4^{vir}$. The following equation can be written:

$$\mathbf{X}_i^{vir} = \sum_{j=1}^4 \alpha_{ij} \mathbf{C}_j^{vir} \quad (7)$$

We expand Equation (5) by Equation (7) and 3D coordinates of each reference point.

$$\lambda \begin{bmatrix} u_i^{vir} \\ v_i^{vir} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \sum_{j=1}^4 \alpha_{ij} \begin{bmatrix} x_j^{vir} \\ y_j^{vir} \\ z_j^{vir} \end{bmatrix} \quad (8)$$

Substituting in the first and two rows, two linear equations can be obtained:

$$\begin{cases} \sum_{j=1}^4 \alpha_{ij} x_j^{vir} - \alpha_{ij} u_i^{vir} z_j^{vir} = 0 \\ \sum_{j=1}^4 \alpha_{ij} y_j^{vir} - \alpha_{ij} v_i^{vir} z_j^{vir} = 0 \end{cases} \quad (9)$$

The unknown parameters of the expression are only related to reference point coordinates. Rewrite Equation (10) to matrix form:

$$\mathbf{M} \mathbf{x} = 0 \quad (10)$$

where \mathbf{M} is a $2n \times 12$ matrix, and $\mathbf{x} = [C_1^{virT} \ C_2^{virT} \ C_3^{virT} \ C_4^{virT}]^T$ is a vector. Vector \mathbf{x} belongs to the right null space of \mathbf{M} matrix. \mathbf{x} is solved by singular value decomposition.

$$\mathbf{x} = \sum_{k=1}^N \beta_k \mathbf{v}_k \quad (11)$$

\mathbf{v}_k is an eigenvector with zero eigenvalues of \mathbf{M} , with scalar coefficients. If N is determined, reference point coordinates can be calculated by Equation (11). We can obtain \mathbf{X}^{vir} in different virtual camera coordinates system by Equation (7). Then, we reproject \mathbf{X}^{vir} to the omnidirectional camera system to obtain the location \mathbf{X}_i^c .

At last, the absolute orientation algorithm [23] is used to directly calculate the rotation matrix \mathbf{R} and translation vector \mathbf{T} of the omnidirectional camera coordinate system relative to the object space coordinate system, that is, the pose of the omnidirectional camera.

$$\mathbf{X}_i^w = \mathbf{R}\mathbf{X}_i^c + \mathbf{T} \quad (12)$$

with $\mathbf{A} = \sum_{i=1}^n \mathbf{X}_i^{wo} (\mathbf{X}_i^{co})^T$.

The relationship between \mathbf{X}_i^{wo} and \mathbf{X}_i^w is shown as Equation (13), and the relationship between \mathbf{X}_i^{co} and \mathbf{X}_i^c is shown in Equation (14).

$$\mathbf{X}_i^{wo} = \mathbf{X}_i^w - \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i^w \quad (13)$$

$$\mathbf{X}_i^{co} = \mathbf{X}_i^c - \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i^c \quad (14)$$

4. Results

4.1. Simulation

We produced synthetic 3D–2D correspondence by a virtual calibrated camera, the intrinsic parameters of which are shown in Table 2. Calibration parameters were selected empirically. Rotation matrix and translation vector were randomly generated. The translation vector norm was distributed into [100, 200]. In the experiment, we generated 10 sets for the input data and kept the number of control points for each setting unchanged.

Table 2. Internal parameters of the camera in the simulation experiment.

Parameter	f_x	f_y	\mathbf{u}_0	\mathbf{v}_0	ζ
Value	260.1	259.6	517.1	385.8	0.97

4.1.1. Synthetic Experiments of Accuracy about Noise

In the first experiment, we assessed the effect of the coordinate error on the accuracy of pose estimation. The Gaussian noise, having zero mean and standard deviation from 0 to 10, was added to the corresponding 2D point coordinates. For each level of standard deviation, we performed 100 experiments. The results of the experiment are shown in Figure 3a,b.

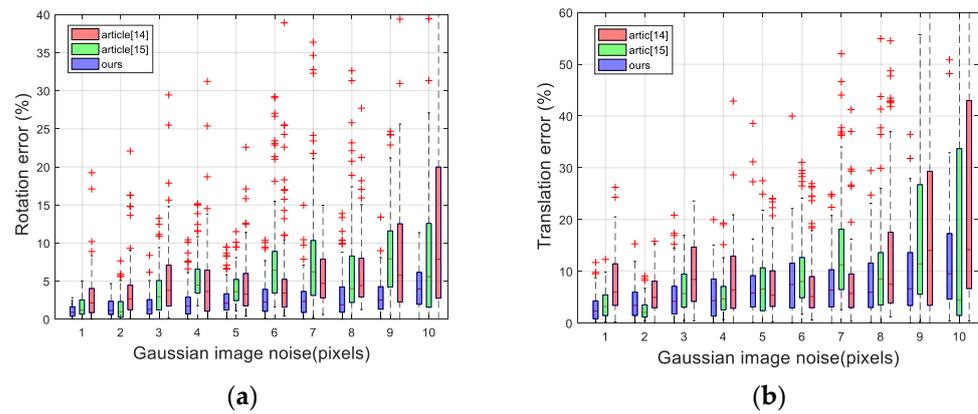


Figure 3. Pose estimation errors for different noise. (a) Rotation matrix error (b) Translation vector error.

We computed the relative error of the estimated rotation by Equations (15) and (16), where \mathbf{q} and $\mathbf{q}_{\text{ttrue}}$ are the normalized quaternion corresponding to the rotation matrix.

$$E_{\text{rot}}(\%) = \|\mathbf{q}_{\text{ttrue}} - \mathbf{q}\| \quad (15)$$

$$E_{\text{trans}}(\%) = \frac{\|\mathbf{t}_{\text{ttrue}} - \mathbf{t}\|}{\|\mathbf{t}\|} \quad (16)$$

In Figure 3a,b, in addition to the rotation error, we also plotted the translation error. The line in the box is the average error. The rectangle represents 50% error distribution. The smaller it is, the more concentrated the error distribution, indicating that the method is more stable. It can be observed that the error of our method grows linearly with the level of noise and remains much lower than all the others. When the noise level is small, the accuracy of the algorithms of [14,15] is similar to that of the algorithm in this paper. When the noise level is large, the error of these two algorithms is large. When the variance of Gaussian noise is 10, the maximum error of the rotation angle is below 10%, the average error of the rotation angle is 3%, and the average error of the translation vector is 12%, which indicates that it has good antinoise performance.

4.1.2. Synthetic Experiments of Accuracy about the Number of Control Points

Simulation experiments were conducted to explore the influence of the number of control points on the pose estimation accuracy. The number of control points on a single surface was varied from 5 to 20, keeping the average noise value at 0 and the standard deviation at 2. The rotation matrix and translation vector were randomly generated, where the norm of the translation vector was [100, 200]. One hundred experiments were performed, and the results are shown in Figure 4a,b.

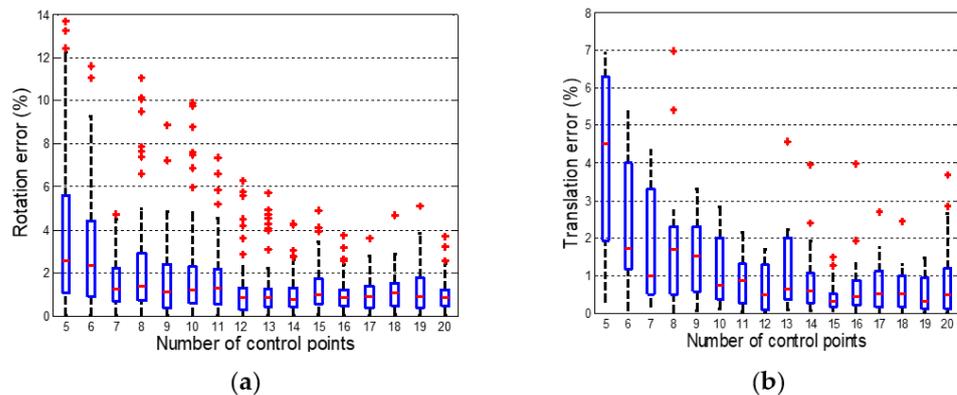


Figure 4. Pose estimation errors for different numbers of control points. (a) Rotation matrix error (b) Translation vector error.

The error decreases as the number of control points increases. Figure 4 shows that the error of the pose estimation is related to the number of control points projected on the virtual planes (plane $Z = -1$ and plane $Z = 1$). When the number of control points is greater than 12, the accuracy approaches stability. When the number of points is less than 6, the pose estimation error increases. For the case where the number of control points is between 6 and 12, the error is uncertain. If the number of control points on each virtual plane does not exceed 6, the error is large. If there are more than 6 control points on a virtual plane, the accuracy is better. The 4 spatial 3D control points have 12 unknown parameters. As shown in Equation (10), two equations are obtained for each control point. Therefore, when the number of control points is greater than or equal to 6, the system has a definite solution.

4.2. Real Images

4.2.1. Calculating Rotation Angle

In the first experiment with real images, we applied the rotating platform to survey the motion matrix of the detection target. The object in the picture was a checkerboard pattern of $10\text{ cm} \times 10\text{ cm} \times 20\text{ cm}$ carved with $30\text{ mm} \times 30\text{ mm}$. The experimental layout is shown in Figure 5. Assuming that ϕ is the rotation angles on the axes of rotation, the rotation matrix \mathbf{R} can be easily obtained by the Rodriguez formula.

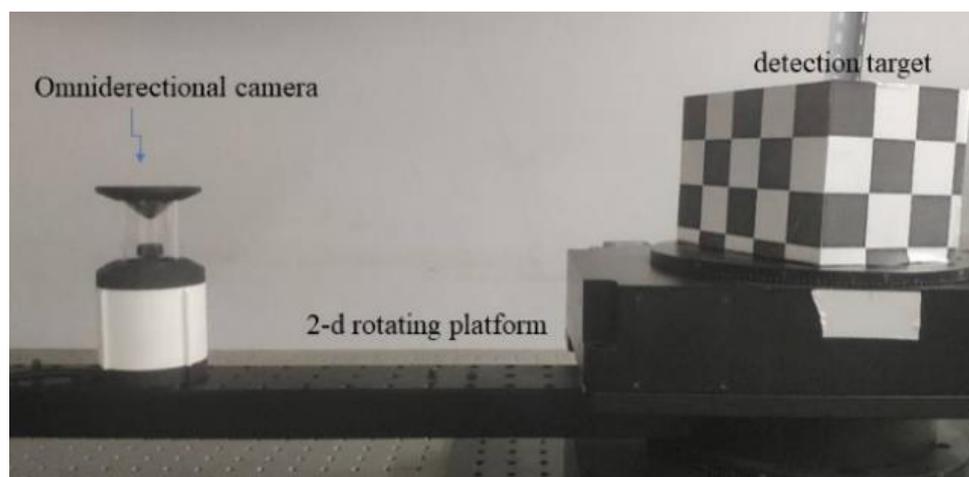


Figure 5. Experimental layout.

We randomly changed the angles ϕ and then captured images. We ignored the difference between the center of the turntable and the origin of the world coordinate system. The coordinates of the checkerboard corners were obtained with the implementation of key point recognition by the method of [24]. This is a gradient-based subpixel intersection detection algorithm. The internal parameter was obtained by running the camera calibration toolbox [22]. The calibration results are shown in Table 3. Then, the pose was estimated by the proposed method and the methods in [14,15].

Table 3. Calibration result.

Parameter	f_x	f_y	u_0	v_0	ζ
Value	370.647	370.018	807.551	597.126	1.027

The pose of the calibration target from camera coordinate system to world coordinate system was calculated before and after exercise, expressed by \mathbf{R}_1 , \mathbf{R}_2 and \mathbf{T}_1 , \mathbf{T}_2 , respectively.

The conversion from the world coordinate system before movement to the world coordinate system after movement is as follows:

$$\begin{aligned} X_{w1} &= R_1^{-1}R_2X_{w2} - R_1^{-1}[T_2 - T_1] \\ &= R_{w1-w2}X_{w2} + T_{w1-w2} \end{aligned} \quad (17)$$

The measured rotation matrix was converted into a form of quaternions and then compared with the measured result. Figure 6 shows calculated the rotation errors and translation errors when the camera position was fixed at 30 mm of the caliper and the rotation angle of the turntable was varied.

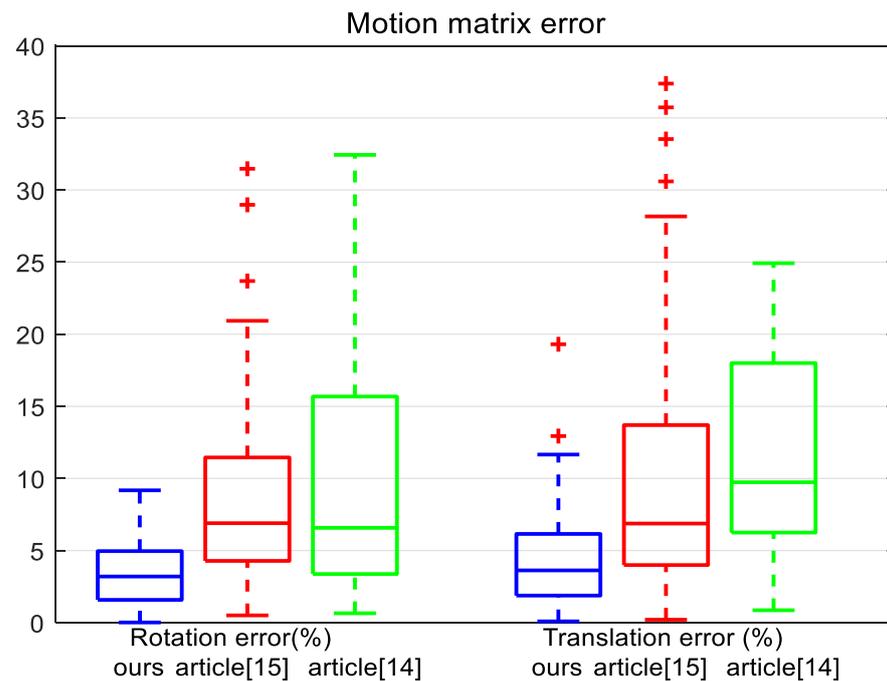


Figure 6. Comparing the accuracy of our approach method, the method of [14], and the method of [15].

In the result of the algorithm of [15], the error distribution is relatively scattered. The line segment representing the average value is in the lower middle position, indicating that most of the most data have small errors and a small part of data have large errors. In the result of the algorithm of [14], there are some large discrete values above the box. This is the fault tolerance of pose calculation due to the plane incline. The results of the proposed algorithm are relatively concentrated. The line segment representing the average value is in the middle of the whole box, which indicates that the distribution of error values is relatively average and the algorithm has good stability. The average value rotation error is about 2.5% and the average value translation error is about 3.0%. Compared with the results of the simulation experiment, both the rotation error and the translation error of the real image experiment become larger. The reason is that the measurement error of camera calibration is more complicated than the simulation error. Rotation angle is calculated from two pictures before and after, which requires higher stability of the algorithm. The error of any picture will affect the calculation of rotation angle.

4.2.2. Reconstruction

In the second experiment, we reconstructed the 3D metric of the object by using two omnidirectional images. In order to study the accuracy of the reconstruction, the object consisted of three orthogonal checkerboard patterns with the size of each square measuring 30 mm × 30 mm, as shown in Figure 7.

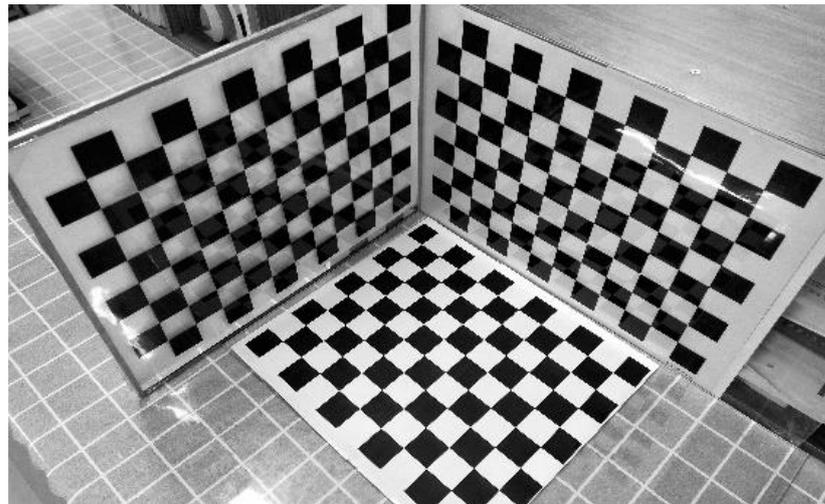


Figure 7. The sample trihedron used for the 3D reconstruction experiment.

We used the omnidirectional camera to take images (see Figure 8) at two different locations. The corner points were manually selected and matched. The pose of the object was estimated by the method proposed in this paper. Then, the relative rotation matrices and translation vectors between the two images were derived. The reconstruction results are shown in Figure 9. In order to compare the reconstructed value with the real value, the least square was used to fit the plane. The angles between the planes were 89.27° , 89.44° , and 89.64° . Finally, we calculated the dimensions of the reconstructed checkerboard. The average error relative to the ideal was 1.85 mm. We also used the methods proposed in [14,15] to conduct experiments. The plane included angles were 95.6° , 85.7° , and 84.5° for the method of [14] and the average error of the corner coordinates was 3.17 mm. The plane included angles were 93.2° , 84.6° , and 83.5° for the method of [15], and the average error of the corner coordinates was 2.65 mm.

The inclination of the plane where the space point is located has a great influence on the pose estimation. When the points are lying on a plane on an ideal plane facing the camera, there is almost nonexistent pose ambiguity, and all the methods have similar accuracy, with almost nonexistent outliers for all the methods. The pose ambiguity problem appears for inclined planes. The reprojection error of the camera pose estimated by the algorithms in [14,15] on plane 1 was small, while the reprojection error on planes 2 and 3 was large.

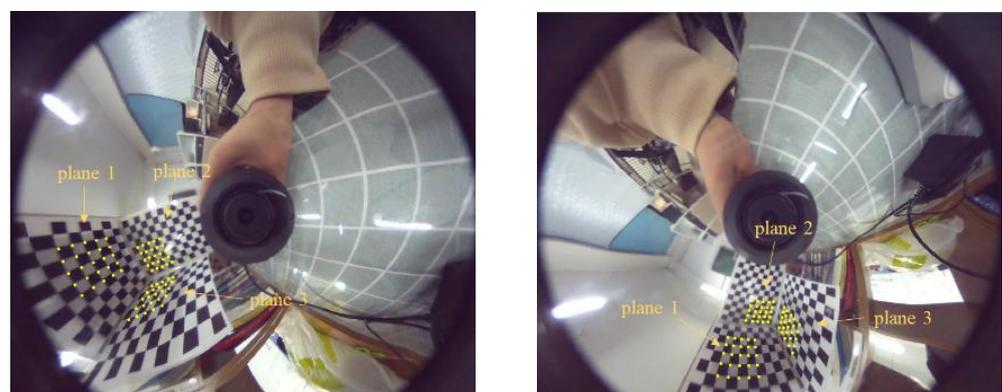


Figure 8. Two pictures of the trihedron taken by the omnidirectional camera. The points used for the 3D reconstruction are indicated by yellow dots.

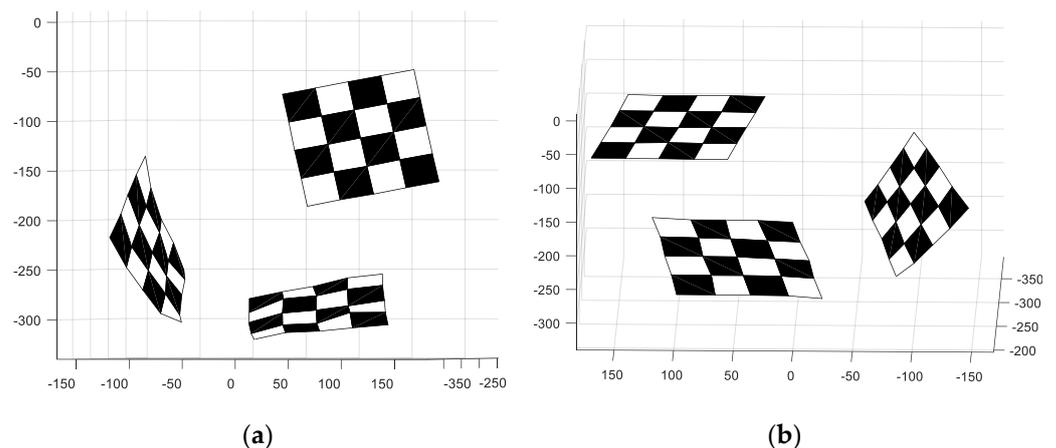


Figure 9. Three rendered views of the reconstructed trihedron. (a,b) are the results of different perspectives.

4.3. Discussion

In this paper, by adding a virtual image plane, the nonlinear imaging process of the omnidirectional camera is transformed into an equivalent linear imaging process. The EPnP algorithm is then applied to the omnidirectional imaging system. The control point coordinates around the camera improve the accuracy of camera pose estimation. A large field of view can effectively reduce the loss of tracking caused by matching. The EPnP algorithm converts the control points into four virtual coordinates to reduce the influence of a single control point on pose estimation, and it has good antinoise performance. In addition, the control point modification of this method only contains the basic mapping relation, and the calculation cost is low. Specifically, the execution time of the coordinate map is about 20 milliseconds per frame, and the execution time of pose estimation is about 10 milliseconds. We think there is room for improvement if the code can be executed on the GPU. However, this method has some limitations in the number of virtual image faces to choose. When more virtual image planes are selected, more control points are needed, and fewer image planes are selected, which leads to larger errors caused by camera approximation. Our plan is to remove these limitations in the future through the adoption of deep learning technologies.

5. Conclusions

In this work, we propose the linear equivalent model for omnidirectional cameras. The omnidirectional camera is equivalent to the combination of two or more virtual cameras. After solving the nonlinear imaging problem of the omnidirectional camera, the EPnP algorithm was extended to the omnidirectional camera. The method can be suitable for all kinds of mirror omnidirectional systems. In the simulation part, we first studied the influence of image point error on pose estimation. Results show that the proposed solutions work well when noise occurs. Then, we investigated the influence of the number of control points on the accuracy of pose estimation, and the accuracy was found to increase with the increase in the number of control points. The four virtual control points in the EPnP algorithm effectively reduce the influence of a single spatial control point on the overall pose estimation. The better antinoise performance of the method was substantiated by simulation and real image experiments. At the same time, space points around the camera can effectively improve the positioning accuracy, which is an advantage brought by the large field of view. In the second experiment of the real image, we put the pose estimation results into the reconstruction algorithm. Three checkerboard calibration plates were reconstructed. We calculated the included angle of the checkerboard calibration plate to evaluate the reconstruction accuracy. In future research, we will extend the EPnP algorithm to the scene reconstruction of the panoramic camera.

In addition, the virtual image plane can solve the problem of large image distortion by retaining the large field of view. The imaging process of the virtual camera is linear. This can speed up the popularization of omnidirectional cameras in a wide range of video surveillance, robot navigation, and other applications in computer vision.

This work has been focused on omnidirectional camera pose estimation. Future work will also be dedicated to location and reconstruction with a large field of view of the system. The powerful tools of deep learning (DL) will also be taken into account for camera positioning by directly processing large field images.

Author Contributions: X.G. and Y.L., methodology; X.G., software, validation; X.G., Y.L. and Y.W., formal analysis; X.G. and M.L., investigation; X.G., writing—original draft preparation; X.G., writing—review and editing; Y.L., supervision; X.X., project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by The National Science Foundation (NSF) for Young Scientists of China (Grant No. 61803045).

Acknowledgments: We are very grateful for the support from College of Optoelectronic Engineering, Changchun University of Science and Technology.

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

References

- Xu, D.; Li, Y.F.; Tan, M. A general recursive linear method and unique solution pattern design for the perspective-n-point problem. *Image Vis. Comput.* **2008**, *26*, 740–750. [[CrossRef](#)]
- 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](#)]
- 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](#)]
- Zhou, B.; Chen, Z.; Liu, Q. An Efficient Solution to the Perspective-n-Point Problem for Camera with Unknown Focal Length. *IEEE Access* **2020**, *8*, 162838–162846. [[CrossRef](#)]
- Meng, C.; Xu, W. ScPnP: A non-iterative scale compensation solution for PnP problems. *Image Vis. Comput.* **2021**, *106*, 104085. [[CrossRef](#)]
- Liu, G.F. Accurate and Robust Monocular SLAM with Omnidirectional Cameras. *Sensors* **2019**, *19*, 4494. [[CrossRef](#)] [[PubMed](#)]
- Patruno, C.; Colella, R.; Nitti, M.; Ren, V.; Mosca, N. A Vision-Based Odometer for Localization of Omnidirectional Indoor Robots. *Sensors* **2020**, *20*, 875. [[CrossRef](#)] [[PubMed](#)]
- Dias, T.; Miraldo, P.; Gon Alves, N. A Framework for Augmented Reality using Non-Central Catadioptric Cameras. *J. Intell. Robot. Syst.* **2016**, *83*, 359–373. [[CrossRef](#)]
- Pais, G.D.; Dias, T.J.; Nascimento, J.C.; Miraldo, P. Omni-DRL: Robust Pedestrian Detection using Deep Reinforcement Learning on Omnidirectional Cameras. In Proceedings of the International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019; pp. 4782–4789.
- Karaim, H.C.; Baris, I.; Bastanlar, Y. Detection and classification of vehicles from omnidirectional videos using multiple silhouettes. *Pattern Anal. Appl.* **2017**, *20*, 893–905. [[CrossRef](#)]
- Morbidi, F.; Caron, G. Phase Correlation for Dense Visual Compass from Omnidirectional Camera-Robot Images. *IEEE Robot. Autom. Lett.* **2017**, *2*, 688–695. [[CrossRef](#)]
- Aliaga, D.G. Accurate Catadioptric Calibration for Real-time Pose Estimation in Room-size Environments. In Proceedings of the IEEE International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; pp. 127–134.
- Paulino, A.; Araujo, H. Pose Estimation for Central Catadioptric Systems: An Analytical Approach. In Proceedings of the Object Recognition Supported by User Interaction for Service Robots, Quebec, QC, Canada, 11–15 August 2002; pp. 696–699.
- Gebken, C.; Tolvanen, A.; Perwass, C. Perspective Pose Estimation from Uncertain Omnidirectional Image Data. In Proceedings of the 18th International Conference, Hong Kong, China, 20–24 August 2006; pp. 793–796.
- Gonçalves, N.; Araújo, H. Linear solution for the pose estimation of noncentral catadioptric systems. In Proceedings of the IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–7.
- Ilizirov, G.; Filin, S. Pose Estimation and Mapping Using Catadioptric Cameras with Spherical Mirrors. *Int. Arch. Photogramm. Remote Sens.* **2016**, *XLI-B3*, 43–47. [[CrossRef](#)]
- Miraldo, P.; Eiras, F.; Ramalingam, S. Analytical Modeling of Vanishing Points and Curves in Catadioptric Cameras. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2012–2018.
- Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An Accurate O(n) Solution to the PnP Problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [[CrossRef](#)]

19. Penatesanchez, A.; Andradecetto, J.; Morenonoguer, F. Exhaustive Linearization for Robust Camera Pose and Focal Length Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2387–2400. [[CrossRef](#)]
20. Deng, F.; Wu, Y.; Hu, Y. Position and Pose Estimation of Spherical Panoramic Image with Improved EPnP Algorithm. *Acta Geod. Cartogr. Sin.* **2013**, *45*, 677–684.
21. Chen, P.; Wang, C. IEPnP: An Iterative Estimation Algorithm for Camera Pose Based on EPnP. *Chin. J. Opt.* **2018**, *38*, 130–136.
22. Mei, C.; Rives, P. Single View Point Omnidirectional Camera Calibration from Planar Grids. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3945–3950.
23. Horn, B.K.; Hilden, H.M.; Negahdaripour, S. Closed-form Solution of Absolute Orientation Using Orthonormal Matrices. *J. Opt. Soc. Am. A* **1988**, *5*, 1127–1135. [[CrossRef](#)]
24. Geiger, A.; Moosmann, F.; Car, O. Automatic camera and range sensor calibration using a single shot. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 3936–3943.