PLOS ONE

RESEARCH ARTICLE

A New Pose Estimation Algorithm Using a Perspective-Ray-Based Scaled Orthographic Projection with Iteration

Pengfei Sun[1], Changku Sun[1], Wenqiang Li[1,2], Peng Wang[1,2]*

1 State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin, China, **2** Science and Technology on Electro-optic Control Laboratory, Luoyang Institute of Electro-optic Equipment, Luoyang, China

* wang_peng@tju.edu.cn

Citation: Sun P, Sun C, Li W, Wang P (2015) A New Pose Estimation Algorithm Using a Perspective-Ray-Based Scaled Orthographic Projection with Iteration. PLoS ONE 10(7): e0134029. doi:10.1371/journal.pone.0134029

Editor: Jonathan A Coles, Glasgow University, UNITED KINGDOM

Received: March 31, 2015

Accepted: July 5, 2015

Published: July 21, 2015

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work is supported by the National Natural Science Foundation of China (No. 51375339), http://www.nsfc.gov.cn/. This work is also supported by the Aviation Science Foundation of China (No. 20135148004). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Pose estimation aims at measuring the position and orientation of a calibrated camera using known image features. The pinhole model is the dominant camera model in this field. However, the imaging precision of this model is not accurate enough for an advanced pose estimation algorithm. In this paper, a new camera model, called incident ray tracking model, is introduced. More importantly, an advanced pose estimation algorithm based on the perspective ray in the new camera model, is proposed. The perspective ray, determined by two positioning points, is an abstract mathematical equivalent of the incident ray. In the proposed pose estimation algorithm, called perspective-ray-based scaled orthographic projection with iteration (PRSOI), an approximate ray-based projection is calculated by a linear system and refined by iteration. Experiments on the PRSOI have been conducted, and the results demonstrate that it is of high accuracy in the six degrees of freedom (DOF) motion. And it outperforms three other state-of-the-art algorithms in terms of accuracy during the contrast experiment.

Introduction

Estimating the pose of a calibrated camera has lots of applications in augmented reality, air refueling, and unmanned aerial vehicle (UAV) navigation [1–3]. The augmented reality often operates on the basis of prior knowledge of the environment, which limits range and accuracy of registration. Pose estimation attempts to locate 3D features in the feature map, and provides registration when the reference map is in the sensing range [4]. In air refueling, a single monocular camera is mounted on the receiver aircraft while the probe and drogue is mounted on the tanker aircraft. Pose estimation algorithm is proposed for the purpose of tracking the drogue during the capture stage of autonomous aerial refueling [5]. In UAV navigation, pose estimation is employed in the formation flying of UAVs. To guarantee the relative positions of these UAVs, the IR-LEDs on the leader UAV is captured by the IR-camera on the follower UAV and the detected features are transmitted to the pose estimation algorithm [6].

Pose estimation, also known in the literature as the Perspective-n-Point (PnP) problem, measures the position and orientation of a calibrated camera with known image features [7]. The features available to solve the PnP problem are usually given in the form of a set of point correspondences, each constituting a space point expressed in object coordinates and its image projection expressed in image coordinates. In the past few decades, a huge amount of work has been done to address the problem. Various solutions to the PnP problem, including the EPnP [8], the DLS [9], the RPnP [10], the ASPnP [11], the LHM [12], etc., are developed. To the best of our knowledge, these PnP solutions can show high accuracy only when dealing with dozens or even hundreds of point correspondences. Unfortunately, considering the terrible environment in pose estimation applications, it is hard to offer too many stable and distinguishable point correspondences. Although the DLS is applicable to situations of $n \leq 7$, the moving range of the target object is extremely limited [9]. The PnP solutions, especially the P4P solutions, have been in great demand in recent years. The P4P solutions can be classified into two types: model-based solutions, which depend on the approximation of a camera model, and geometric configuration solutions that handle the relationship between image space and object space with geometrical characteristic such as distance, angle, parallel, vertical, etc.. POSIT is a popular solution to the non-coplanar P4P problem and is one of the representative solutions in the first category [13]. Scaled orthographic projection is employed in the algorithm, and the rotation matrix and translation vector of a calibrated camera is obtained through the projection. Iteration is also introduced to refresh the old image coordinates of feature points, and then repeat the previous steps. The iteration does not stop until the output has satisfied the preset accuracy or the algorithm is circulated for preset times. For the stability and high accuracy, the POSIT is continuously introduced into applications in complex interference environment [14–19]. The latter solutions take advantage of the geometric configuration of the special feature points. The geometric configuration of the P4P problem is a core research. Liu. M. L. et al. [20] made full use of the geometric configuration of the four non-coplanar feature points, including the angle between two perspective lines, the mixed product among the perspective lines, the segments in object space, etc.. The follow-up researches did not surpass the category of the geometric configuration by Liu. M. L.. Z. Y. Hu et al. [21] mathematically analyzed the geometric configuration of non-coplanar P4P problem. They parameterized the relationship between the numbers of possible solutions and the numbers of geometric configuration. Wu PC et al. [22] focused on the plausible pose, and proposed an analytical motion model to interpret, or even eliminate, the geometric illusion. Yang Guo [23] researched the coplanar P4P problem. By converting perspective transformation to affine transformation and using invariance to 3D affine transformation, it is found that the upper bound of the coplanar P4P problem is two. A technique based on a singular value decomposition (SVD) is also proposed for the coplanar P4P problem by Yang Guo, unverified by any real test. To improve estimation accuracy, Long Li et al. [24] introduced Frobenius norm into the determinant of rotation matrix, instead of the SVD-based method. Unfortunately, the proposed method did not contribute to accuracy and noise resistance, it only reduced the runtime. Bujnak M. et al. [25] and Kuang Y. et al. [26] focused on the recovery of the unknown focal length from the P4P solutions, and were not interested in the accuracy of the P4P solutions. From the studies in [13–26], it can be concluded that the research concerning accuracy improvement of the P4P solutions is slow and unattractive.

To sum up, the camera model of the above solutions is a pinhole camera, in which all the incident rays are projected directly onto the detector plane through a single point, called the effective pinhole of the camera model [27]. In practice, the incident rays are deviated on account of the compound lenses. The P4P solutions are negatively influenced by the imprecise camera model. There are still other expressions proposed to describe the camera model [28–31]. By using lens geometry model, the geometric relationship between images and objects is

established via Snell's Law and skew ray tracking in [28] and [29]. The camera model is represented by a matrix equation that relates the parameters of the image plane with the incident ray. But it is complex because each incident ray is represented by a set of six pose parameters. In a general imaging model, the cameral is regarded as "black box" [30, 31]. A set of virtual sensing elements called "raxel" is used to describe a linear mapping from incident rays to the image plane. The "raxel" is composed of three parameters: an image projection, the yaw and pitch directions of the projective ray. The calibration of "raxel" is tedious and entirely depends on the accuracy of rotation stage. In this paper, an incident ray tracking model (IRT) is proposed, where two reference planes are regarded as camera model parameters. Through analyzing the geometric properties of the proposed model, the incident ray is mathematically summarized as a perspective ray which is positioned by two points respectively located in the two reference planes. Considering the excellent scaled orthographic projection, a perspective-ray-based scaled orthographic projection is employed in this paper. The projection formulates a linear system which calculates the approximation of object pose, and iteration loops are also introduced to obtain a more accurate approximation. The camera calibration based on the incident ray tracking model (IRT) and the perspective-ray-based scaled orthographic projection with iteration (PRSOI) for pose estimation will be described in detail in the following sections.

Incident Ray Tracking Model

Incident ray formulation

Fig 1 shows the imaging system, helpful to formulate the mathematical model of imaging sensors. Irrespective of its specific design, the purpose of an imaging system is to map incident rays from the scene onto pixels on the detector.

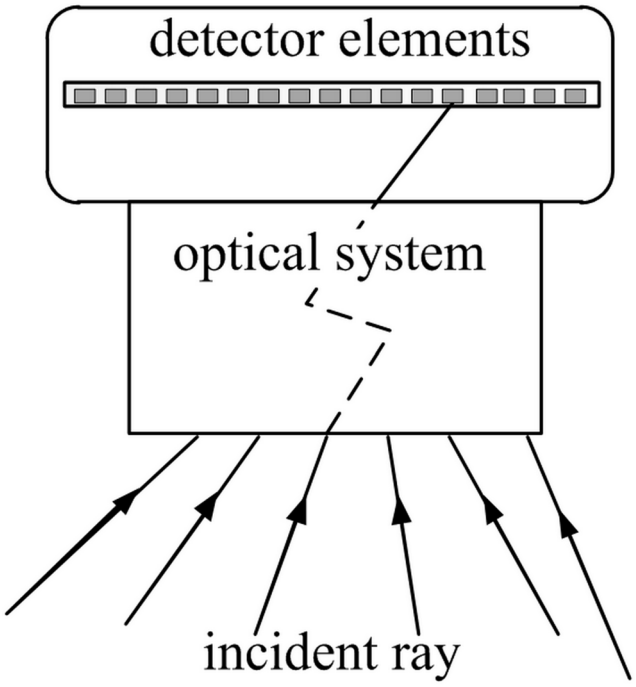

Fig 1. An incident ray passing through an imaging system which is absorbed by detector elements (pixel).

doi:10.1371/journal.pone.0134029.g001

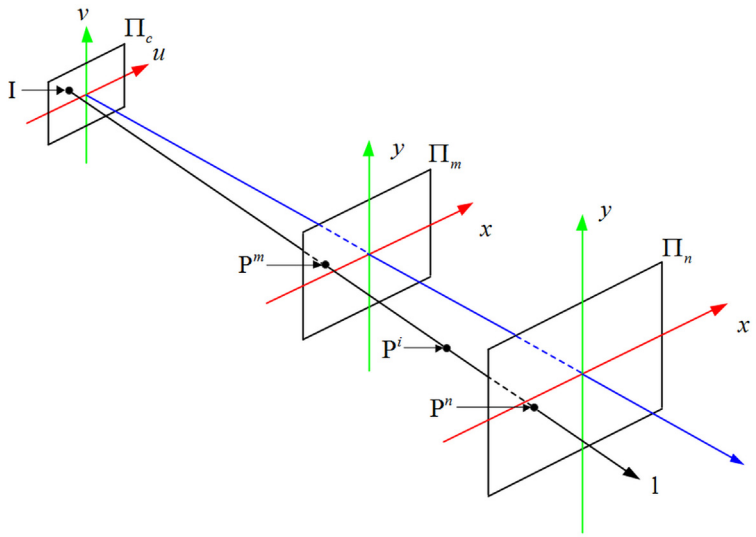

Fig 2. Geometrization of the IRT.

doi:10.1371/journal.pone.0134029.g002

Each pixel in Fig 1 collects energy from the incident ray in the optical system that has a non-zero aperture size. However, the incident ray can be represented by a perspective ray when studying the geometric properties of the imaging system. As shown in Fig 1, the system maps the incident ray to the pixel. Because the path that incident ray traverses from scene to the pixel can be arbitrarily complex, the incident ray should be replaced by an abstract mathematical equivalent that is referred to as a perspective ray l (I, P^m, P^n). The IRT is composed of the incident rays, in the field of view. In the following section, the parameters of the IRT will be introduced.

Parameters of camera model

If the radiometric response function of each perspective ray is computable, one can linearize the radiometric response with respect to the image plane. In our context of camera model, the ray to image mapping may be parameterized as Fig 2.

A point $P^i(x,y,t)$ imaged at (x,y) at depth t is imaged along a perspective ray l (t is the vertical distance from P^i to the Π_n). It will be more convenient to represent the model if the perspective rays, such as l, are arranged on two planes called reference planes, such as Π_m and Π_n. Each perspective ray will intersect the two reference planes respectively at only one point, namely P^m and P^n. The reference planes could be written as a function:

$$\begin{cases} \Pi_m(x,y) = \{P^m\} \\ \Pi_n(x,y) = \{P^n\} \end{cases} \quad (1)$$

A perspective ray could be determined uniquely through the reference planes Π_m and Π_n, and the IRT is parameterized by the two reference planes.

Computing parameters

The parameters used to specify the IRT are derived from the reference planes. The perspective ray passes through the two reference planes $\Pi_m(x, y)$ and $\Pi_n(x, y)$, and intersects the image

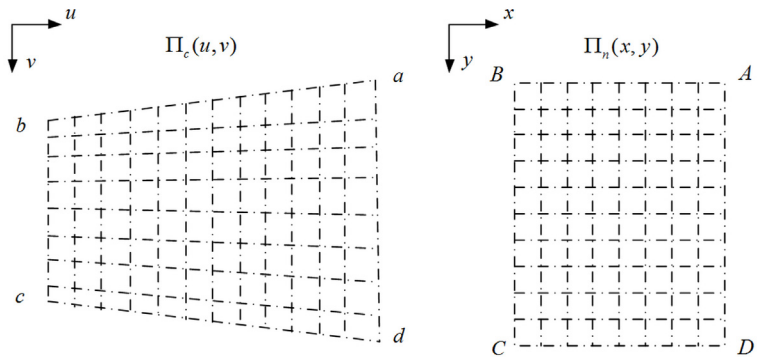

Fig 3. A mapping between reference plane and image plane. The intersection points of dotted lines in plane ABCD corresponds to the ones in plane abcd.

doi:10.1371/journal.pone.0134029.g003

plane $\Pi_c(u, v)$ at point I(u,v). Ignoring the position of the planes, the mapping from $\Pi_n(x, y)$ to $\Pi_c(u, v)$ is represented as Fig 3.

There is a one-to-one mapping between the image plane and the reference plane. As the two planes are both represented by a set of points, the mapping is recasted as the following equation:

$$\begin{cases} x = \sum_{i=0}^{n}\sum_{j=0}^{n-i} C_{ij} u^i v^j \\ y = \sum_{i=0}^{n}\sum_{j=0}^{n-i} D_{ij} u^i v^j \end{cases} \tag{2}$$

where (C_{ij},D_{ij}) are the mapping parameters, n is the order of the mapping, (x,y) is the space coordinates of the points in the plane ABCD while (u,v) is the image coordinates of them in the plane abcd.

The (C_{ij},D_{ij}) are obtained using Levenberg-Marquard method [32]. The reference plane $\Pi_m(x, y)$ is represented by rational functions $g_x^m(u, v)$ and $g_y^m(u, v)$, consisting of C_{ij}^m and D_{ij}^m. m can be replaced by n.

Pose Estimation Based on Perspective Ray

Object pose formulation

Considering the geometrical features of the perspective ray $l_k(I_k, P_k^m, P_k^n)$, which is described in Fig 4. The points P_0^i and P_k^j are located on the object. $P_0^i-\mathbf{uvw}$ is the object coordinate system, and O-**ijk** is the reference plane coordinate system.

Pose estimation in this paper aims to compute the rotation matrix and translation vector of the object. The purpose of the rotation matrix is to transform the object coordinates such as $\overrightarrow{P_0^i P_k^j}$ into coordinates defined in the reference plane coordinate system such as $\overrightarrow{P_0^n P_k^n}$ (n represents a point located on the plane Π_n). The dot product $\overrightarrow{P_0^i P_k^j} \bullet \mathbf{i}$ between the vector $\overrightarrow{P_0^i P_k^j}$ and the first row of the matrix correctly provides the projection of this vector on the unit vector **i** of

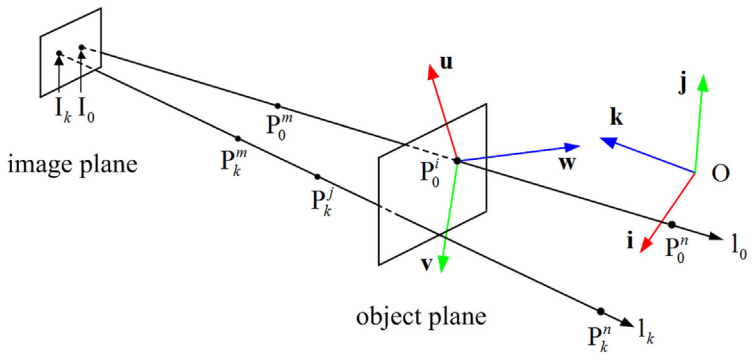

Fig 4. The perspective rays used for object pose.

doi:10.1371/journal.pone.0134029.g004

the reference plane coordinate system. The rotation matrix can therefore be written as:

$$\mathbf{R} = \begin{bmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{bmatrix} \tag{3}$$

where i_u, i_v, i_w are the coordinates of **i** in the object coordinate system. To compute the rotation matrix, it is only needed to compute **i** and **j** in the object coordinate system. The vector **k** is then obtained by the cross-product **i** × **j**.

The translation vector, **T**, is the vector $\overrightarrow{OP_0^i}$. The point P_0^i is determined by the perspective ray l_0 which can be expressed as:

$$\begin{cases} f_x^0(z) = g_x^n(u_0, v_0) + (g_x^m(u_0, v_0) - g_x^n(u_0, v_0))(z - z^n)/(z^m - z^n) \\ f_y^0(z) = g_y^n(u_0, v_0) + (g_y^m(u_0, v_0) - g_y^n(u_0, v_0))(z - z^n)/(z^m - z^n) \end{cases} \tag{4}$$

where z^m and z^n are respectively the z coordinate of the planes Π_m and Π_n. From Eq (4), the vector $\overrightarrow{OP_0^i}$ could be expressed as:

$$\overrightarrow{OP_0^i} = (f_x^0(z^i) - g_x^n(0,0), f_y^0(z^i) - g_y^n(0,0), z^i - z^n) \tag{5}$$

where z^i is the z coordinate of the plane Π_i. Therefore to compute the object translation, only the z coordinate needs computing. Thus the object pose is fully defined once the unknowns **i**, **j** and z are found.

Projection on the perspective ray

The image point corresponding to the feature point, which projects on the perspective ray, is shown in Fig 5. Only two feature points P_0^i and P_k^j appeared in the projection. The perspective rays l_0 and l_k are respectively in correspondence with the feature points P_0^i and P_k^j, and are computed by the calibration parameters. The object coordinate system is centered at P_0^i, and the coordinate of P_k^j relative to P_0^i is known. The point P_0^i locates on the plane Π_i which parallels to the planes Π_m and Π_n.

Scaled orthographic projection is an approximation to the perspective projection. It is assumed that the depths of different points can all be set as the same depth z^i. The geometric construction to obtain the perspective ray l_k of P_k^j in a perspective projection and the

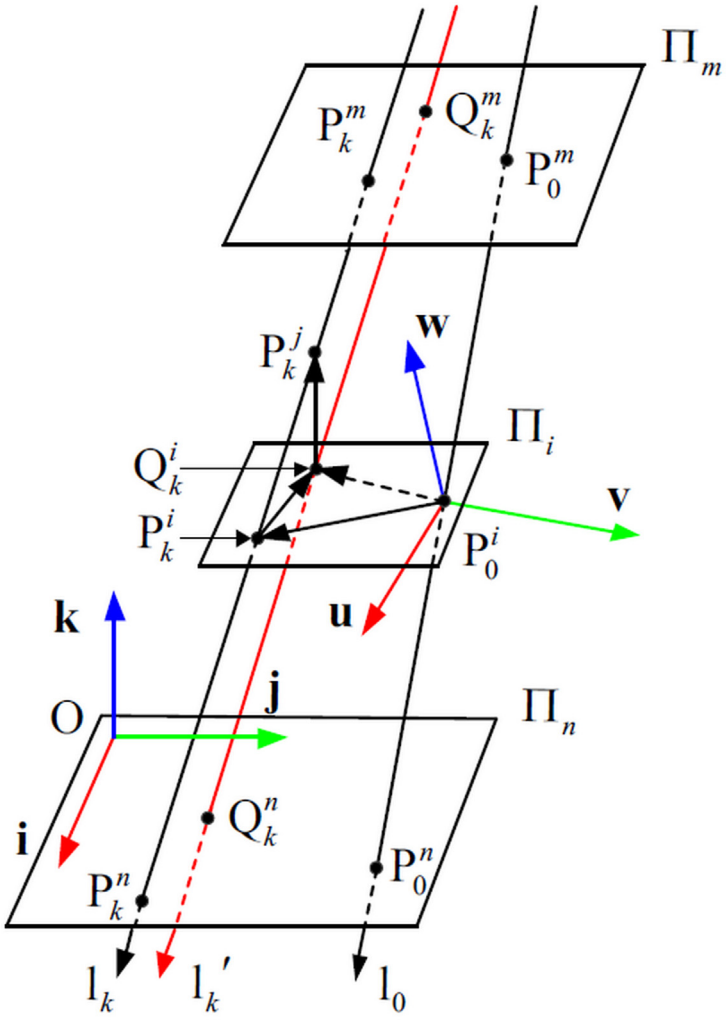

Fig 5. An imaging model of the perspective rays. It includes a perspective projection and a scaled orthographic projection.

doi:10.1371/journal.pone.0134029.g005

perspective ray $1_k'$ of P_k^j in a scaled orthographic projection is shown in Fig 5. The point P_k^j is projected on the plane Π_i at Q_k^i by a scaled orthographic projection.

Formulations of projections

Formulations of perspective projection. Now consider the equations that characterize a perspective projection and relate the unknown row vectors **i** and **j** of the rotation matrix and the unknown z^j coordinate of the translation vector to the known coordinates of the vector $\overrightarrow{P_0^i P_k^j}$ in the object coordinate system, and to the known coordinates of P_0^n. In Fig 6, the perspective ray l_k intersects the plane Π_i in P_k^i, and P_k^j projects on the plane Π_i at Q_k^i. The vector $\overrightarrow{P_0^i P_k^j}$ is the sum of three vectors:

$$\overrightarrow{P_0^i P_k^j} = \overrightarrow{P_0^i P_k^i} + \overrightarrow{P_k^i Q_k^i} + \overrightarrow{Q_k^i P_k^j} \quad (6)$$

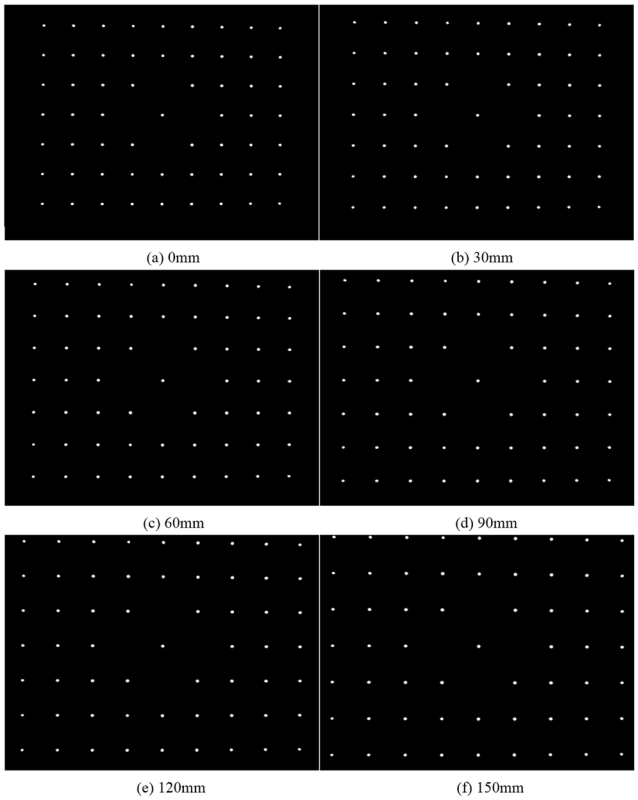

Fig 6. Captured images at six positions.

doi:10.1371/journal.pone.0134029.g006

The vector $\overrightarrow{P_0^i P_k^i}$ is constrained by two perspective rays l_0 and l_k. It can be expressed as:

$$\overrightarrow{P_0^i P_k^i} = (f_x^k(z^i) - f_x^0(z^i), f_y^k(z^i) - f_y^0(z^i), 0) \tag{7}$$

where (f_x^0, f_y^0) and (f_x^k, f_y^k) are the functions of l_0 and l_k. The vector $\overrightarrow{P_k^i Q_k^i}$ is also constrained by l_k and l_k'. For the z coordinate of P_k^j is $z^{i\prime} = z^i(1+\varepsilon^i)$ ($\varepsilon^i = \overrightarrow{P_0^i P_k^j} \bullet \mathbf{k}/z^i$), the vector $\overrightarrow{P_k^i Q_k^i}$ is defined as:

$$\overrightarrow{P_k^i Q_k^i} = (f_x^k(z^{i\prime}) - f_x^k(z^i), f_y^k(z^{i\prime}) - f_y^k(z^i), 0) \tag{8}$$

The vector $\overrightarrow{Q_k^i P_k^j}$ is perpendicular to the reference plane Π_i, and it can be defined as:

$$\overrightarrow{Q_k^i P_k^j} = (0, 0, z^i \cdot \varepsilon^i) \tag{9}$$

The sum of the three vectors can then be expressed as:

$$\overrightarrow{P_0^i P_k^j} = (f_x^k(z^{i\prime}) - f_x^0(z^i), f_y^k(z^{i\prime}) - f_y^0(z^i), z^i \cdot \varepsilon^i) \tag{10}$$

Then take the dot product of Eq (10) with the unit vector **i** and **j**. The dot products $\overrightarrow{P_0^i P_k^j} \bullet \mathbf{i}$

and $\overrightarrow{P_0^i P_k^j} \bullet \mathbf{j}$ are expressed as:

$$\begin{cases} \overrightarrow{P_0^i P_k^j} \cdot \mathbf{i} = f_x^k(z^{i\prime}) - f_x^0(z^i) \\ \overrightarrow{P_0^i P_k^j} \cdot \mathbf{j} = f_y^k(z^{i\prime}) - f_y^0(z^i) \end{cases} \tag{11}$$

Solving Eq (11) for the unknowns would provide all the information required to define the object pose.

Formulations of scaled orthographic projection. The right hand sides of Eq (11), the terms $f_x^k(z^{i\prime})$ and $f_y^k(z^{i\prime})$, are in fact the coordinates of the point Q_k^i, which are the scaled orthographic projections of the feature point P_k^j. Consider the points P_0^i, P_k^j, and the projections Q_k^i of P_k^j on the plane Π_i, the vector $\overrightarrow{P_0^i P_k^j}$ is the sum of two vectors $\overrightarrow{P_0^i Q_k^i}$ and $\overrightarrow{Q_k^i P_k^j}$. The vector $\overrightarrow{P_0^i Q_k^i}$ should be represented as:

$$\overrightarrow{P_0^i Q_k^i} = (f_x^k(z^{i\prime}) - f_x^0(z^i), f_y^k(z^{i\prime}) - f_y^0(z^i), 0) \tag{12}$$

Then take the dot product of the vector $\overrightarrow{P_0^i P_k^j}$ with the unit vector $\mathbf{i}$. The dot product $\overrightarrow{Q_k^i P_k^j} \bullet \mathbf{i}$ is zero, and the dot product $\overrightarrow{P_0^i Q_k^i} \bullet \mathbf{i}$ is the x coordinate $f_x^k(z^{i\prime}) - f_x^0(z^i)$. Consequently, the dot products $\overrightarrow{P_0^i P_k^j} \bullet \mathbf{i}$ and $\overrightarrow{P_0^i P_k^j} \bullet \mathbf{j}$ are similar to Eq (11).

Iteration for scaled orthographic projection

Eq (11) can also be written:

$$\begin{cases} \overrightarrow{P_0^i P_k^j} \cdot \mathbf{i} = f_x^k(z^i(1+\varepsilon^i)) - f_x^0(z^i) \\ \overrightarrow{P_0^i P_k^j} \cdot \mathbf{j} = f_y^k(z^i(1+\varepsilon^i)) - f_y^0(z^i) \end{cases} \tag{13}$$

As the points P_0^i and P_0^n, P_k^j and P_k^n respectively locate in the perspective rays l_0 and l_k, Eq (13) could be approximated as:

$$\begin{cases} \overrightarrow{P_0^i P_k^j} \cdot \mathbf{I} = f_x^k(z^n) - f_x^0(z^n) \\ \overrightarrow{P_0^i P_k^j} \cdot \mathbf{J} = f_y^k(z^n) - f_y^0(z^n) \end{cases} \tag{14}$$

where $\mathbf{I} = s_i \cdot \mathbf{i}$, $\mathbf{j} = s_i \cdot \mathbf{j}$. Eq (14) provides a linear system of equations in which the only unknowns are respectively the coordinates of $\mathbf{I}$ and $\mathbf{J}$. The norm of $\mathbf{I}$ and $\mathbf{J}$ are respectively the scaling factor s_i and s_j between the vector $\overrightarrow{P_0^i P_k^j}$ and $\overrightarrow{P_0^n P_k^n}$. Then the length of the two vectors can be written as:

$$|\overrightarrow{P_0^n P_k^n}| = \frac{s_i + s_j}{2} |\overrightarrow{P_0^i P_k^j}| \tag{15}$$

It can be parameterized as:

$$|\overrightarrow{P_0^n P_k^n}| = \frac{s_i + s_j}{2} \sqrt{(f_x^k(z^{i\prime}) - f_x^0(z^i))^2 + (f_y^k(z^{i\prime}) - f_y^0(z^i))^2 + (z^i \cdot \varepsilon^i)^2} \tag{16}$$

If values are given to the term ε^i, z^i is obtained from Eq (16).

The proposed algorithm, used to determine the pose by solving the linear system, is called perspective-ray-based scaled orthographic projection (PRSO). The solution of the PRSO

algorithm is only an approximation if the values given to the term ε^i are not exact. But once the unknowns **i** and **j** have been computed, more exact values can be computed for the term ε^i, and the equations can be solved again with these better values. The iteration algorithm is named PRSOI (PRSO with Iterations). It generally makes the values of **i**, **j** and z^i converge towards values which correspond to a correct pose through iterations.

Initially, the term ε^i is equal to zero. In fact, it can be assumed that P_k^j and Q_k^i coincide. When tracking an object, the initial value for the term ε^i is preferably chosen equal to the value obtained at the last iteration of the pose estimation for the previous image. The computed error of coordinates, which is between the projection point Q_k^n of P_k^j in the prior iteration and the one in the current iteration, reaches the minimum at the end of iterations.

Solving the system of PRSO algorithm

Within the preceding iterative algorithm, the solution of Eq (14) is still a problem. This equation could be rewritten in a more compact form:

$$\begin{cases} \overrightarrow{P_0^i P_k^j} \cdot \mathbf{I} = \xi^i \\ \overrightarrow{P_0^i P_k^j} \cdot \mathbf{J} = \eta^i \end{cases} \tag{17}$$

where $\xi^i = f_x^k(z^n) - f_x^0(z^n), \eta^i = f_y^k(z^n) - f_y^0(z^n)$. The dot products of this equation are expressed in terms of vector coordinates in the object coordinate frame:

$$\begin{cases} [u_i \quad v_i \quad w_i][i_u \quad i_v \quad i_w]^T = \xi^i \\ [u_i \quad v_i \quad w_i][j_u \quad j_v \quad j_w]^T = \eta^i \end{cases} \tag{18}$$

These are linear equations where the unknowns are the coordinates of **I** and **J**. The other parameters are known: $f_x^0, f_y^0, f_x^k, f_y^k$ are the known functions of l_0 and l_k, and u_i, v_i, w_i are the known coordinates of P_k^j in the object coordinate frame. Substitute the n feature points for Eq (18), a linear system is generated for the coordinates of the unknown vectors **I** and **J**:

$$\begin{cases} \mathbf{A} \cdot \mathbf{I} = \mathbf{x}' \\ \mathbf{A} \cdot \mathbf{J} = \mathbf{y}' \end{cases} \tag{19}$$

where **A** is the matrix of the coordinates of the object points in the object coordinate frame, $\mathbf{x}' = [\xi_0^i \cdots \xi_j^i \cdots \xi_n^i]^T$, $\mathbf{y}' = [\eta_0^i \cdots \eta_j^i \cdots \eta_n^i]^T$. In general, if there are at least four non-coplanar points, the least square solution of the linear system is given by:

$$\begin{cases} \mathbf{I} = \mathbf{B} \cdot \mathbf{x}' \\ \mathbf{J} = \mathbf{B} \cdot \mathbf{y}' \end{cases} \tag{20}$$

where the object matrix **B** is the pseudo inverse of the matrix **A**. Once the least square solutions to **I** and **J** are obtained, the unit vectors **i** and **j** are simply obtained by normalizing **I** and **J**.

Now the translation vector **T** of the object can be obtained. It is vector $\overrightarrow{OP_0^i}$, and z^i is computed by Eq (16). Then the vector **T** is computed by Eq (5).

Experiment Results

Camera calibration results

In the experiment, a domestically developed CCD camera with image resolution 768×576 pixels, pixel size 0.0083mm×0.0086mm, and field angle 60°, is used. It is fixed on a linear stage via a bracket. The type of the linear stage is Zolix KSA300-11-X, with repeatability of 3μm, straightness of 10μm, and travel of 300mm. The calibration target is a solid circular array pattern with 7×9 circular points evenly distributed. The size of the target is 500×600mm^2, and the distance between the adjacent points is 60mm in the horizontal and vertical directions.

Fix the target on the optical platform, and then move the camera to make the calibration target cover most of the field of view. The captured images are taken at six different positions, and the distance between the adjacent positions is 30mm. Two specific images, such as the images captured at 0mm and 150mm, are regarded as the calibration data. As the camera parameters are obtained, the captured images, including the two specific ones, are introduced into the IRT to compute the space error of the calibration points. The captured images are shown in Fig 6.

Table 1 lists the camera parameters. The reference planes Π_m and Π_n are described as the fifth order polynomials.

Fig 7 shows the position distribution of the calibration points. The standard position is the standard coordinates of the calibration points, and the calculated position is the calculated coordinates of the calibration points obtained by the IRT and the image coordinates from the captured images.

Table 1. Camera parameters.

Parameters	Π_m(0mm)		Π_n(150mm)	
	$g_x^m(\mathbf{u},\mathbf{v})$	$g_y^m(\mathbf{u},\mathbf{v})$	$g_x^n(\mathbf{u},\mathbf{v})$	$g_y^n(\mathbf{u},\mathbf{v})$
(C_{00}, D_{00})	-2.654E+02	-1.865E+02	-3.226E+02	-2.208E+02
(C_{10}, D_{10})	6.726E-01	-4.893E-04	8.756E-01	-8.562E-03
(C_{01}, D_{01})	2.479E-02	7.093E-01	2.791E-03	8.171E-01
(C_{20}, D_{20})	9.571E-05	-1.246E-05	-2.688E-04	2.292E-05
(C_{11}, D_{11})	-1.179E-04	-1.156E-04	1.512E-04	8.542E-05
(C_{02}, D_{02})	1.432E-06	-1.468E-05	-6.406E-05	8.410E-05
(C_{30}, D_{30})	-3.328E-07	2.558E-08	5.994E-07	-9.413E-08
(C_{21}, D_{21})	4.067E-07	1.902E-07	-6.830E-07	-3.960E-07
(C_{12}, D_{12})	-2.716E-08	5.363E-08	-8.002E-08	-2.445E-07
(C_{03}, D_{03})	-6.565E-08	-8.220E-09	2.443E-07	-1.878E-07
(C_{40}, D_{40})	5.180E-10	-6.346E-11	-5.699E-10	1.449E-10
(C_{31}, D_{31})	-7.111E-10	-8.815E-12	1.226E-09	8.289E-10
(C_{22}, D_{22})	-2.782E-12	1.190E-10	-4.663E-10	1.900E-10
(C_{13}, D_{13})	-3.097E-11	-1.939E-10	3.740E-10	8.478E-11
(C_{04}, D_{04})	2.582E-10	3.799E-11	-1.669E-11	5.031E-10
(C_{50}, D_{50})	-2.645E-13	4.620E-14	2.081E-13	-9.186E-14
(C_{41}, D_{41})	4.766E-13	-1.756E-13	-7.830E-13	-6.358E-13
(C_{32}, D_{32})	4.283E-14	-1.214E-13	3.333E-13	-5.590E-14
(C_{23}, D_{23})	-1.650E-13	1.326E-13	-1.334E-13	-1.493E-13
(C_{14}, D_{14})	-5.200E-14	-2.771E-13	2.536E-13	-3.852E-13
(C_{05}, D_{05})	1.835E-14	3.582E-13	-5.181E-13	2.097E-13

doi:10.1371/journal.pone.0134029.t001

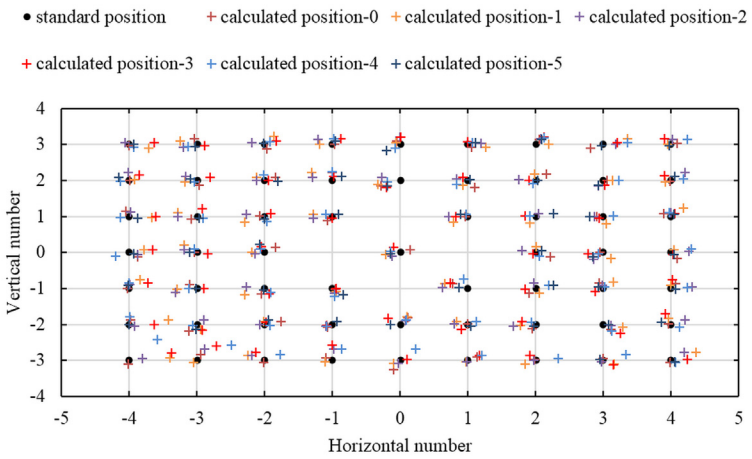

Fig 7. Position distribution of the calibration points. The "•" represents the standard two-dimensional coordinate of the calibration points while the "+" represents the calculated two-dimensional coordinate of them.

doi:10.1371/journal.pone.0134029.g007

The root mean square error (RMSE) of the calculated calibration points is 0.17mm in horizontal direction, and 0.12mm in vertical direction. According to the error statistics of the calibration points, it is obvious that the camera can be described by the IRT completely.

Pose estimation results

The experiment devices for pose estimation are shown in Fig 8. The integrated rotation stage is composed of three rotation stages: Zolix RAK-200 in the yaw direction, Zolix RAK-100 in the pitch and roll directions. The repeatability of the RAK-200 is 0.005°, load 50kg. The repeatability of the RAk-100 is 0.005°, load 30kg. The type of interface controller is Zolix MC600-4B, two-phase stepping motor, closed-loop control. The codes of the P4P solutions are run in Microsoft Visual Studio 2010 environment on a computer with 3.40 GHz CPU.

During the experimental process, the target is fixed on the rotating platform, and the image is captured at every 1°. The rotating angle of the target between the initial position and the current position is measured by the two captured images. The three directions of rotational motion are tested. Then fix the target on the linear stage, and capture the image of it at every

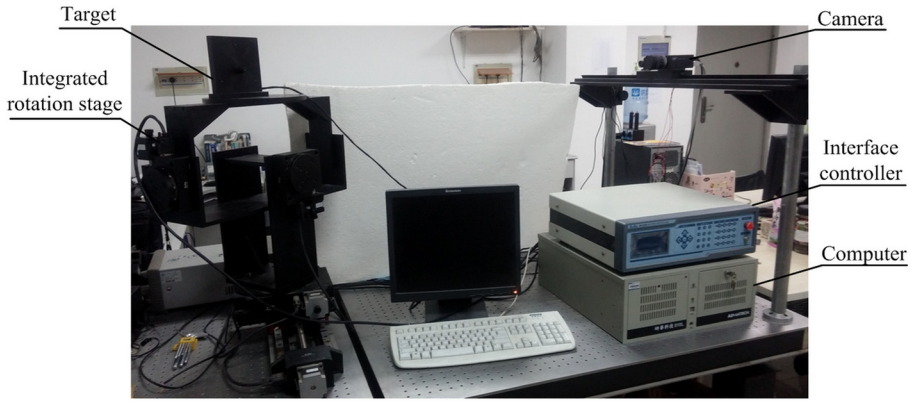

Fig 8. Experiment devices.

doi:10.1371/journal.pone.0134029.g008

Fig 9. A sample image used for pose estimation.

doi:10.1371/journal.pone.0134029.g009

2mm. The moving distance of the target between the initial position and the current position is also measured by the two captured images. The three directions of translational motion are tested. Fig 9 shows a real image of four non-coplanar feature points captured by the calibrated camera.

Notice the central part in the Fig 9: the effective coverage of the four feature points in the captured image is just about 1.49%. This is very different from the captured image of the

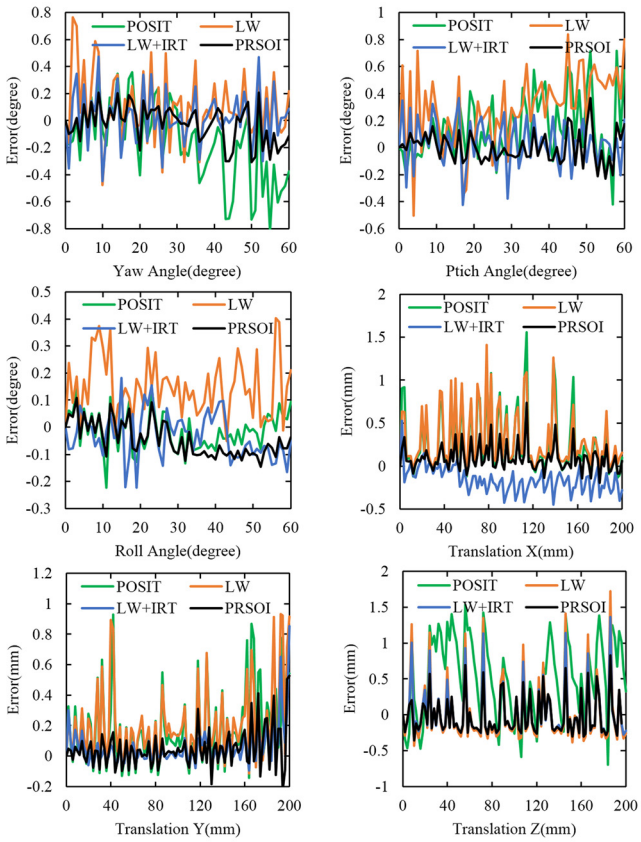

Fig 10. Pose estimation error distribution.

doi:10.1371/journal.pone.0134029.g010

Table 2. The RMSE of the PnP algorithms.

method	e_{ry}[a](*deg.*)	e_{rp}[b](*deg.*)	e_{rr}[c](*deg.*)	e_{tx}[d](*mm*)	e_{ty}[e](*mm*)	e_{tz}[f](*mm*)
POSIT	0.290	0.243	0.071	0.369	0.241	0.552
LW	0.258	0.277	0.110	0.340	0.248	0.448
LW+IRT	0.201	0.176	0.083	0.146	0.130	0.362
PRSOI	0.136	0.115	0.062	0.152	0.128	0.272

[a]ry is rotation in yaw direction,
[b]rp is rotation in pitch direction,
[c]rr is rotation in roll direction,
[d]tx is translation in x direction,
[e]ty is translation in y direction, and
[f]tz is translation in z direction.

doi:10.1371/journal.pone.0134029.t002

popular P*n*P solutions. The PRSOI is tested by the captured data, and compared with the state-of-the-art P4P solutions. For the pinhole camera, the geometric configuration solution by Liu ML and Wong KH [20], denoted by LW in short, as well as the popular iterative solution POSIT [13], are considered. For the IRT camera, the LW+IRT solution is considered, since the LW incorporates the IRT. The results of the P4P solutions are shown in Fig 10. The calculated pose of the target are checked by comparison with the standard positions which are obtained from the interface controller.

Statistics are used in estimation error analysis, and the RMSE of the P4P solutions are summarized in Table 2.

Through the comparison between the LW and the LW+IRT, it is obvious that the accuracy of the LW+IRT is higher than that of the LW. The result suggested that the IRT is effective in the P4P solutions. As the accuracy of the PRSOI is higher than that of the POSIT, it demonstrates that the perspective-ray-based scaled orthographic projection is superior to the scaled orthographic projection in a pinhole camera. Considering the accuracy of the four P4P solutions, it can be proved that the accuracy of the PRSOI outperforms the other three state-of-the-art P4P solutions.

The PRSOI is an iterative solution, though powerful, does have a shortfall: planning the correct pose for each position is slow. In this paper, accuracy is the major concern while computational cost is ignored.

Conclusion

This paper puts forward and deeply analyzes the IRT and the PRSOI. The IRT, which with definite geometric meaning, consists of two reference planes Π_m and Π_n. The PRSOI introduces the IRT into a scaled orthographic projection, then adopts an iteration to make the perspective-ray-based scaled orthographic projection more accurate. Four non-coplanar points are used as feature points in the real image experiment. And three other P4P solutions are introduced to be compared with the PRSOI. Experiment results demonstrated that the PRSOI is of high accuracy in the six-DOF motion. The P4P solution proposed in this paper is of significance in the P4P applications such as the positioning of mechanical arm, the four-wheel aligners, the installation of super-huge workpiece, etc..

To the best of our knowledge, it is the first study to incorporate the perspective ray with the scaled orthographic projection, and the incorporation works effectively in the P4P situation.

Supporting Information

S1 Dataset. Camera captured dataset. This archive contains the captured data files used as the basis for the P4P solutions described in the manuscript. The data are provided in a directory hierarchy where each degree of freedom has a separate directory. And the calibration data is the captured data used in the camera calibration.
(ZIP)

Author Contributions

Conceived and designed the experiments: PS. Performed the experiments: PS PeW. Analyzed the data: WL PW. Contributed reagents/materials/analysis tools: CS. Wrote the paper: PS CS PW.

References

1. Li B, Mu C, Wu B. A survey of vision based autonomous aerial refueling for Unmanned Aerial Vehicles. ICICIP 2012: Third Int Conf Intelligent Control and Information; 2012; IEEE; 2012. 1–6.
2. Xu G, Qi X, Zeng Q, Tian Y, Guo R, Wang B. Use of land's cooperative object to estimate UAV's pose for autonomous landing. Chinese J Aeronaut. 2013; 26(6): 1498–1505.
3. Taketomi T, Okada K, Yamamoto G, Miyazaki J, Kato H. Camera pose estimation under dynamic intrinsic parameter change for augmented reality. Computers & Graphics. 2014; 44(0): 11–19.
4. Klein G, Murray D. Parallel tracking and mapping for small AR workspaces. ISMAR 2007: 6th IEEE and ACM Int Symp on Mixed and Augmented Reality; 2007; IEEE; 2007. 225–234.
5. Martínez C, Richardson T, Thomas P, Du Bois JL, Campoy P. A vision-based strategy for autonomous aerial refueling tasks. Robot Auton Syst. 2013; 61(8): 876–895.
6. Park J-S, Lee D, Jeon B, Bang H. Robust vision-based pose estimation for relative navigation of unmanned aerial vehicles. ICCAS 2013: 13th Int Conf Control, Automation and Systems; 2013; IEEE; 2013. 386–390.
7. Lepetit V, Fua P. Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. Foundations and Trends in Computer Graphics and Vision. 2005; 1(1): 1–89.
8. Lepetit V, Moreno-Noguer F, Fua P. EPnP: An Accurate O(n) Solution to the PnP Problem. Int J Comput Vision. 2009; 81(2): 155–166.
9. Hesch JA, Roumeliotis SI. A Direct Least-Squares (DLS) method for PnP. ICCV 2011: Proceeding of the 13th International Conference on Computer Vision; 2011; Barcelona, Spain. IEEE; 2011. 383–390.
10. Shiqi L, Chi X, Ming X. A Robust O(n) Solution to the Perspective-n-Point Problem. IEEE Trans Pattern Analysis and Machine Intelligence. 2012; 34(7): 1444–1450.
11. Zheng Y, Sugimoto S, Okutomi M. ASPnP: An Accurate and Scalable Solution to the Perspective-n-Point Problem. IEICE Trans Inf & Syst. 2013; 96(7): 1525–1535.
12. Lu CP, Hager GD, Mjolsness E. Fast and globally convergent pose estimation from video images. IEEE Trans Pattern Anal Mach Intell. 2000; 22(6): 610–622.
13. Dementhon D, Davis L. Model-based object pose in 25 lines of code. Int J Comput Vision. 1995; 15(1–2): 123–141.
14. David P, DeMenthon D, Duraiswami R, Samet H. SoftPOSIT: Simultaneous Pose and Correspondence Determination. Int J Comput Vision. 2004; 59(3): 259–284.
15. Gramegna T, Venturino L, Cicirelli G, Attolico G, Distante A. Optimization of the POSIT algorithm for indoor autonomous navigation. Robot Auton Syst. 2004; 48(2): 145–162.
16. He M, Ratanasawanya C, Mehrandezh M, Paranjape R. UAV Pose Estimation using POSIT Algorithm. International Journal of Digital Content Technology and its Applications. 2011; 5(4): 153–159.
17. Martins P, Batista J. Monocular Head Pose Estimation. In: Campilho A. and Kamel M., editors. Image Analysis and Recognition. Springer Berlin Heidelberg; 2008.
18. Bertók K, Sajó, Levente,Fazekas, Attila. A robust head pose estimation method based on POSIT algorithm. Argumentum. 2011.
19. Woo Won K, Sangheon P, Jinkyu H, Sangyoun L. Automatic head pose estimation from a single camera using projective geometry. Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on; 2011; 2011. 1–5.

20. Liu ML, Wong KH. Pose estimation using four corresponding points. Pattern Recogn Lett. 1999; 20(1): 69–74.
21. Hu ZY, Wu FC. A Note on the Number of Solutions of the Noncoplanar P4P Problem. IEEE Trans Pattern Analysis and Machine Intelligence. 2002; 24(4): 550–555.
22. Wu P-C, Tsai Y-H, Chien S-Y. Stable pose tracking from a planar target with an analytical motion model in real-time applications. MMSP 2014: IEEE 16th Int Workshop on Multimedia Signal Processing; 2014; IEEE; 2014. 1–6.
23. Yang G. A note on the number of solutions of the coplanar P4P problem. Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on; 2012; 2012. 1413–1418.
24. Li L, Deng Z-Q, Li B, Wu X. Fast vision-based pose estimation iterative algorithm. Optik—International Journal for Light and Electron Optics. 2013; 124(12): 1116–1121.
25. Bujnak MK, Z; Pajdla, T. A general solution to the P4P problem for camera with unknown focal length. CVPR 2008: IEEE Conference on Computer Vision and Pattern Recognition. 2008; 1–8.
26. Kuang Y, Astrom K. Pose Estimation with Unknown Focal Length Using Points, Directions and Lines. 2013; 529–536.
27. Sturm P, Ramalingam S, Tardif J-P, Gasparini S, Barreto J. Camera models and fundamental concepts used in geometric computer vision. Foundations and Trends® in Computer Graphics and Vision. 2011; 6(1–2): 1–183.
28. Lin PD, Sung C-K. Camera calibration based on Snell's law. J DYN SYST-T ASME. 2006; 128(3): 548–557.
29. Lin PD, Sung C-K. Matrix-based paraxial skew ray-tracing in 3D systems with non-coplanar optical axis. Optik. 2006; 117(7): 329–340.
30. Grossberg MD, Nayar SK. A general imaging model and a method for finding its parameters. ICCV 2001: IEEE Int Conf Computer Vision; 2001; Vancouver, Canada. IEEE; 2001. 108–115 vol.102.
31. Grossberg MD, Nayar SK. The raxel imaging model and ray-based calibration. Int J Comput Vision. 2005; 61(2): 119–137.
32. Marquardt DW. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics. 1963; 11(2): 431–441.