Closed-form camera pose and plane parameters estimation for moments-based visual servoing of planar objects

Yuhan Chen¹, Xiao Luo², Baoling Han¹, Jianfeng Jiang³ and Yang Liu¹

Abstract
Image moments are global descriptors of an image and can be used to achieve control-decoupling properties in visual servoing. However, only a few methods completely decouple the control. This study introduces a novel camera pose estimation method, which is a closed-form solution, based on the image moments of planar objects. Traditional position-based visual servoing estimates the pose of a camera relative to an object, but the pose estimation method directly estimates the pose of an initial camera relative to a desired camera. Because the estimation method is based on plane parameters, a plane parameters estimation method based on the 2D rotation, 2D translation, and scale invariant moments is also proposed. A completely decoupled position-based visual servoing control scheme from the two estimation methods above was adopted. The new scheme exhibited asymptotic stability when the object plane was in the camera field of view. Simulation results demonstrated the effectiveness of the two estimation methods and the advantages of the visual servo control scheme compared with the classical method.

Keywords
Pose estimation, image moments, visual servoing, planar object

Introduction
Visual servoing control refers to the use of computer vision data to control the motion of a robot. In visual servoing, two closely linked problem themes are subjects of active research: the design of visual features pertinent to the robotic task to be realized and a control scheme with the chosen visual features such that the desired characteristics are obtained during visual servoing. The latter adopts the control scheme of ensuring an exponential decoupled decrease in error. The former employs the observed parameters of the visual feature in the image of geometric primitives (points, straight lines, ellipses, and cylinders) in image-based visual servoing (IBVS), In position-based visual servoing (PBVS), the geometric primitives reconstruct the camera pose, which then serves as input for visual servoing. The above approaches subject the image stream to an ensemble of measurement processes,
including image processing, image matching, and visual tracking steps, from which the visual features are determined. With large measurement errors, PBVS can be affected by instability in pose estimation and IBVS designed from the image points is subject to local minima, singularity, and limited convergence domain. This is due to the image mismatch, strong nonlinearities, and coupling in the interaction matrix of image points. As a solution to these issues, image moments were introduced for visual servoing.

Image moments have a broad spectrum of applications in image analysis, such as invariant pattern recognition, pose estimation, and reconstruction. A set of moments computed from a digital image represents the global characteristics of the image shape and provides much information regarding the different geometrical features of the image. Because the descriptors used are global and nongeometrical, they avoid image processing such as feature extraction, matching, and tracking. Moments are also useful in achieving control-decoupling properties and choosing a minimal number of features to control the whole degrees of freedom of a camera. Therefore, visual servoing using image moments obtains a large convergence domain and adequate robot trajectories, because of the reduced nonlinearities and coupling in interaction matrix of the adequate combinations of moments.

The analytical form of the interaction matrix related to any moment generated from segmented images was determined, and the result was applied to classical geometric primitives. However, this method requires an object plane configured such that the object and camera planes are parallel to the desired position. Another drawback is the strong coupling of the interaction matrix. Tahri and Chaumette designed a decoupled control scheme to weaken the coupling of the visual features and functions when the object is parallel to the image plane. A generalization of the aforementioned property, that is, when the desired object position is unparallel to the image plane, has achieved excellent experimental results, but it should determine what virtual rotation to apply to the camera and that the x- and y-axes angular velocities of the camera are still coupled. To solve the problem of selecting the image moments to control the rotational motions around the x- and y-axes simultaneously with the translational motions along the same axis, new visual features computed from a low-order shifted moment invariant are proposed.

On the basis of the shifted moments, the selection of a unique feature vector independent of the object shape is proposed and exploited in IBVS. Although this method significantly enlarges the convergence domain of the closed-loop system, the interaction matrix is also coupled. The above methods are all IBVS schemes. Presently, few studies on PBVS control schemes based on image moments exist, but the hybrid visual servoing is very popular. A novel moment-based 2 1/2D visual servoing method for grasping textureless planar parts is proposed by He et al. Instead of applying high-order image moments, it uses rotation features, providing a decoupled interaction matrix that has a full rank and has no local minimum in the control scheme. However, the estimation method of the relative rotation of the textureless parts in real time is based on the cross-correlation analysis and is not a closed-form solution.

To handle the above issues more effectively, this study proposes a closed-form solution of camera pose estimation based on the image moments of planar objects. This method directly estimates the relative pose of the initial camera and the desired camera and not the pose of the camera relative to the object plane. Because the estimation method follows plane parameters, a plane parameters estimation method based on invariant moments is also proposed. Finally, we adopted the PBVS control scheme, which has a completely decoupled interaction matrix.

The rest of the article is organized as follows. The second section discusses preliminary knowledge; the third section introduces camera pose estimation based on image moments; the fourth section is devoted to plane parameters estimation based on the invariant moments; the fifth section discusses the PBVS control scheme and stability analysis; the sixth section presents the simulation results obtained from the co-simulation of MATLAB and CoppeliaSim; and finally, the seventh section outlines the conclusions and future work.

**Preliminaries**

This section introduces the preliminary knowledge used in this study. The pinhole camera model and imaging geometry are introduced, which play important roles in deriving formulas. Then, the relationships between the image moments of two cameras are described, laying the foundation for two camera pose estimation.

**The pinhole camera model and imaging geometry**

From the pinhole camera model, for a 3D coordinate \( \mathbf{P} = [X, Y, Z]^T \) in the camera frame, which projects images as 2D homogeneous normalized coordinates \( \mathbf{p} = [x, y, 1]^T \), we have \( \mathbf{p} = \mathbf{P}/Z \). The image plane measurement (in pixels) of points is given by \( \mathbf{p} = [p_x, p_y, 1]^T \). The relationship between \( \mathbf{p} \) and \( \mathbf{p} \) is expressed as \( \mathbf{p} = \mathbf{K}^{-1} \mathbf{p} \), where \( \mathbf{K} \) is a nonsingular matrix containing the intrinsic parameters of the camera. To simplify the notation, we will assume that any quantity is expressed in the normalized space. This is equivalent to assuming a calibrated camera, that is, full knowledge of the calibration matrix \( \mathbf{K} \).

The homogeneous normalized coordinates of the two camera image points in Figure 1(a) are \( \mathbf{p} = [x, y, 1]^T \), \( \mathbf{p} = [x, y, 1]^T \), satisfying a certain relationship. Assuming that point \( \mathbf{p} \) on planar \( b \) is represented as \( \mathbf{P} \) and \( \mathbf{P} \) in...
the camera frames $F_o$ and $F_n$, respectively, then $^n p$ and $^o p$ satisfy
\[
^n Z^n p = ^n P
\]
\[
^o Z^o p = ^o R_n ^n P + ^o t_n
\]
where $^n Z$ and $^o Z$ are the depth of the observed point $^n P$ relative to the camera frames $F_o$ and $F_n$, respectively, and $^o R_n$ and $^o t_n$ are the rotation matrix and translation vector of the camera frame $F_n$ relative to the camera frame $F_o$, respectively. Substituting for $^n Z^n p$ using equation (1), equation (2) can be written as
\[
^o p = \frac{^n Z^n p}{^o Z} ^o R_n ^n p + \frac{^o t_n}{^o Z}
\]

The following will only consider a planar or has a planar limb surface. Here, the depth $Z$ of any object plane can be expressed as a continuous function of its image coordinates (see the study of Espiau and Chaumette\cite{espiau1994} for more details) and as follows
\[
\frac{1}{Z} = A x + B y + C
\]
where $A$, $B$, and $C$ are the object plane parameters, and we define $N = [A B C]^T$. The parameters of a plane containing typical primitives has been studied by Espiau and Chaumette\cite{espiau1994} and Chaumette et al.\cite{chaumette1994} To find the parameters of a plane containing any pattern, the general method for estimating plane parameters based on the invariant moments will be introduced in the fourth section.

**Image moments**

Moments are generic (and intuitive) descriptors computed from several kinds of objects defined either from closed contours or from a set of points. The $(p + q)^{th}$ order $2D$ geometric moments are denoted by $m_{pq}$ and can be expressed as
\[
m_{pq} = \int_\zeta x^p y^q f(x,y) dydxdy, \quad \forall p, q \in \mathbb{N}
\]
where $\zeta$ is the region of the normalized space in which the image intensity function $f(x,y)$ is defined. The intensity centroid $(x_g, y_g)$ is given by
\[
x_g = \frac{m_{10}}{m_{00}}, \quad y_g = \frac{m_{01}}{m_{00}}
\]

The moments computed with respect to the intensity centroid are called centered moments and are defined as
\[
c_{pq} = \int_\zeta (x - x_g)^p(y - y_g)^q f(x,y) dydxdy
\]
The following considers objects defined by a binary region. Therefore, we assume that $f(x_i, y_j) = 1$ in all the region that defines the object.

In the studies of Espiau and Chaumette\cite{espiau1994} and Chaumette,\cite{chaumette1994} the interaction matrix $L_{c_{pq}}$ related to any centered moment defined from equation (7) has been determined. So we can get
\[
\hat{c}_{pq} = L_{c_{pq}} v_c
\]
where the relative kinematic screw $v_c = [v_c \omega_c]^T$ between the camera and the object, where $v_c$ and $\omega_c$ represent the translational and rotational velocity components, respectively. And the interaction matrix $L_{c_{pq}}$ can be calculated as
\[
L_{c_{pq}} = [c_{vx} \quad c_{vy} \quad c_{vz} \quad c_{wx} \quad c_{wy} \quad c_{wz}]
\]
where

\[
\begin{align*}
c_{vx} &= -(p + 1)Ac_{pq} - pBc_{p-1,q+1} \\
c_{vy} &= -qAc_{p+1,q-1} - (q + 1)Bc_{pq} \\
c_{vz} &= -Ac_{vx} + Bc_{wx} + (p + q + 2)Cc_{pq} \\
c_{wx} &= (p + q + 3)c_{p+1,q+1} + px_{Cp+1,q+1} - 4pn_{11}c_{p-1,q} \\
&\quad + (p + 2q + 3)y_{c_{pq}} - 4qn_{02}c_{pq+1} \\
c_{wy} &= -(p + q + 3)c_{p+1,q} - (2p + q + 3)x_{c_{pq}} \\
&\quad - qy_{c_{pq+1}} + 4pn_{20}c_{p-1,q} + 4qn_{11}c_{pq+1} \\
c_{wz} &= pC_{p-1,q+1} - qC_{p-1,q+1}
\end{align*}
\]

where \( n_{pq} \) is defined as \( n_{pq} = c_{pq}/c_{00} \).

**Relationship between the image moments of two cameras**

If two cameras are not parallel to the object plane (see Figure 1(a)), then \( ^oZ \) and \( ^aZ \) in equation (3) are not constants and they will complicate the calculation of the image moment. Therefore, we introduce a simple situation where two cameras are parallel to the object plane (see Figure 1(b)).

We denote the initial and transformed image coordinates of the feature pattern by \((^oX, ^oY)\) and \((^aX, ^aY)\), respectively, the corresponding intensity functions by \(f(^oX, ^oY)\) and \(g(^aX, ^aY)\), and their image moments by \(m_{pq}^o\) and \(m_{pq}^a\). Therefore

\[
om_{pq}^o = \int_{\zeta} \left( ^oX \right) \left( ^oY \right) \left( ^oX \right) \left( ^oY \right) \int_{\theta} \left( ^oX \right) \left( ^oY \right) \int_{\gamma} \left( ^oX \right) \left( ^oY \right)(9)
\]

Assuming that the image intensity values are preserved during the transformation, we have \(f(^oX, ^oY) = g(^aX, ^aY)\). Further, \(d(^oX)\left( ^oY \right) = \Delta(d(^oX))\left( ^oY \right)\), where \(\Delta\) is the Jacobian of the transformation. The following notations are used for the moments of the initial and transformed images

\[
\begin{align*}
\mu_{pq} &= \frac{o^\prime m_{pq}}{o^\prime m_{00}}, & \nu_{pq} &= \frac{o^\prime m_{pq}}{o^\prime m_{00}}
\end{align*}
\]

According to equation (3), when the two cameras are parallel to the object plane (see Figure 1(b)), we can obtain the following

\[
\begin{bmatrix}
^oX \\
^oY
\end{bmatrix}
= \begin{bmatrix}
&A_{11} &A_{12} \\
&A_{21} &A_{22}
\end{bmatrix}
\begin{bmatrix}
^aX \\
^aY
\end{bmatrix}
+ \begin{bmatrix}
\mu_{01} \\
\mu_{02}
\end{bmatrix}
\]

(11)

where \( A = ^oZ^aZ \), \(^oZ\) and \(^aZ\) are the depths of the observed point \(^oP\) relative to the camera frames \( F_{o} \) and \( F_{a} \), respectively. The expressions relating to the image moments can be derived by substituting the image coordinate transformation equation (11) and different values of \( p \) and \( q \) in equation (9). The moment equations up to second order are given below

\[
\begin{align*}
o^\prime m_{00} &= A(A_{11}A_{22} - A_{12}A_{21})^a \mu_{00} \\
o_{10} &= A(A_{11}v_{10} + A_{12}v_{01})^a \mu_{01} \\
o_{01} &= A(A_{21}v_{10} + A_{22}v_{01})^a \mu_{02} \\
o_{20} &= A^2(A_{11} \mu_{10} + A_{12} \mu_{01} + A_{22} \mu_{02})^a \mu_{02} \\
o_{11} &= A^2(A_{11} \mu_{10} + A_{12} \mu_{01})^a \mu_{11}
\end{align*}
\]

(12)

**Camera pose estimation based on image moments**

This part estimates the relative homogeneous transformation of the two camera frames based on the image moments of the object plane in the two cameras and the plane parameters. In other words, the rotation matrix \(^aR_a\) and the translation vector \(^aT_a\) in Figure 1(a) will be estimated using image moments and plane parameters. The estimation method is a closed-form solution.

**Estimate the rotation matrix of the two camera frames**

The estimation method of the rotation matrix \(^aR_a\) in Figure 1(a) will be introduced in this section. Firstly, to make the imaging plane of the camera parallel to the object plane, the camera frames \( F_a \) and \( F_o \) need to be rotated, and the rotation matrices are \(^oR_o\) and \(^oR_o\), respectively. In other words, the camera frames \( F_a \) and \( F_o \) in Figure 1(a) will be converted to the camera frames \( F_{o'}\) and \( F_{a'}\) in Figure 1(b). Then, the rotation matrix \(^aR_a\) of camera frame \( F_{o'}\) relative to \( F_{o'}\) in Figure 1(b) will be calculated according to the image moment. Finally, the rotation matrix \(^aR_a\) can be calculated as

\[
^aR_a = ^aR_o^T \cdot ^aR_a \cdot ^aR_o
\]

(13)

Any rotation matrix \( R \) can be expressed by \( Z - Y - X \) Euler angles as follows

\[
R = R_z(\gamma) \cdot R_y(\beta) \cdot R_x(\alpha)
\]

where \( R_z(\cdot), R_y(\cdot), \) and \( R_x(\cdot) \) are rotation operations about coordinate frame axes \( z, y, \) and \( x \), respectively, and \( c_{\alpha} = \cos(\alpha), s_{\alpha} = \sin(\alpha), c_{\beta} = \cos(\beta), s_{\beta} = \sin(\beta), \)
The calculation of the rotation matrices $^{a}R_{a}$ and $^{o}R_{o}$. From equation (4), the formula of the object plane relative to the camera frame can be expressed as $AX + BY + CZ = 0$. Therefore, the normal vector of the object plane in Figure 1(a) is expressed as $^{n}N = [nA nB nC]^T$ and $^{o}N = [oA oB oC]^T$ in camera frames $F_{n}$ and $F_{o}$, respectively. $^{a}N$ and $^{o}N$ are normalized as $^{n}n = [nA nB nC]^T$ and $^{o}n = [oA oB oC]^T$.

To make the imaging plane of the camera parallel to the object plane, the camera frame $F_{o}$ is rotated (see Figure 1(a)). The Euler angles of this rotation matrix $^{a}R_{a}$ are $^{a}\gamma = 0, ^{a}\beta, ^{a}\alpha$, so we can obtain

$$^{a}R_{a} = R_{z}(0) \cdot R_{y}^{(a)}(\beta) \cdot R_{z}(^{a}\alpha)$$  

(14)

We find that the unit normal vector of the object plane in the rotated frame $F_{o}$ is $^{n}n = [0 0 1]^T$. Therefore, we can get

$$^{n}n = ^{o}R_{o} \cdot ^{n}n$$  

(15)

To calculate Euler angles $^{a}\beta$ and $^{a}\alpha$, equation (15) is expressed as

$$\begin{align*}
^{a}ac_{\gamma} + ^{a}bs_{\gamma} + ^{a}cc_{\gamma} &= 0 \\
^{a}bc_{\alpha} - ^{a}cs_{\alpha} &= 0 \\
^{a}as_{\beta} + ^{a}bs_{\gamma}c_{\beta} + ^{a}cc_{\gamma}c_{\beta} &= 1
\end{align*}$$  

(16)

As a result, the solution of equation (16) is obtained

$$\begin{align*}
^{a}\beta &= \text{atan2}(s_{\gamma}, c_{\beta}) \\
^{a}\alpha &= \text{atan2}(s_{\alpha}, c_{\alpha})
\end{align*}$$  

(17)

where

$$\begin{align*}
s_{\alpha} &= \pm \sqrt{\frac{n}{b^2 + n c^2}} \\
c_{\alpha} &= \frac{n}{b} s_{\alpha} \\
s_{\beta} &= -\frac{n}{a} c_{\alpha} \\
c_{\beta} &= \frac{n}{b} s_{\alpha} + \frac{n}{c} c_{\alpha}
\end{align*}$$

Similarly, the camera frame $F_{o}$ in Figure 1(a) can also be rotated. The Euler angles of this rotation matrix $^{o}R_{o}$ are $^{o}\gamma = 0, ^{o}\beta, ^{o}\alpha$, so we can also get

$$^{o}R_{o} = R_{z}(0) \cdot R_{y}^{(o)}(\beta) \cdot R_{z}(^{o}\alpha)$$  

(18)

Therefore, the Euler angles $^{o}\beta$ and $^{o}\alpha$ can be obtained by analogy to equation (17).

The calculation of the rotation matrix $^{a}R_{a}$. Firstly, the images $n$ and $o$ are rotated to $n'$ and $o'$ by $^{a}R_{a}$ and $^{o}R_{o}$, respectively. Then, the following shows how to estimate the rotation matrix $^{a}R_{a}$ of frame $F_{a}$ relative to frame $F_{o}$ based on the image moments.

Because the $z$ axes of frames $F_{n}$ and $F_{o}$ are parallel to each other (see Figure 1(b)), $^{a}R_{a}$ can be calculated by the image-centered moments. The rotation matrix $^{a}R_{a}$ can be calculated by

$$^{a}R_{a} = R_{z}(\Delta \gamma)$$  

(19)

where $\Delta \gamma = ^{o}\gamma - ^{n}\gamma$. $^{o}\gamma$ and $^{o}\gamma$ are the orientation angles of the inertial principal axes of the target plane $b$ on the images of cameras $n'$ and $o'$. The orientation angles $^{o}\gamma$ and $^{o}\gamma$ can be calculated as (see the study of Horn24 for more details)

$$\begin{align*}
\sin(2\gamma) &= \frac{2c_{11}}{\sqrt{4c_{11}^2 + (c_{20} - c_{02})^2}} \\
\cos(2\gamma) &= \frac{c_{20} - c_{02}}{\sqrt{4c_{11}^2 + (c_{20} - c_{02})^2}}
\end{align*}$$  

(20)

where $c_{11}$, $c_{20}$, and $c_{02}$ are the second-order central moments and are defined by equation (7). It is well known that $\sin(2\Delta \gamma)$ and $\cos(2\Delta \gamma)$ can be expressed as

$$\begin{align*}
\sin(2\Delta \gamma) &= \sin(2^{o}\gamma) \cos(2^{n}\gamma) - \cos(2^{o}\gamma) \sin(2^{n}\gamma) \\
\cos(2\Delta \gamma) &= \cos(2^{o}\gamma) \cos(2^{n}\gamma) + \sin(2^{o}\gamma) \sin(2^{n}\gamma)
\end{align*}$$  

(21)

It is easy to get that

$$\begin{align*}
2\sin(\Delta \gamma) \cos(\Delta \gamma) &= \sin(2\Delta \gamma) \\
2\cos^2(\Delta \gamma) - 1 &= \cos(2\Delta \gamma)
\end{align*}$$  

(22)

Thus, the solution of equation (22) is obtained

$$\Delta \gamma = \text{atan2}\left(\sin(\Delta \gamma), \cos(\Delta \gamma)\right)$$  

(23)

where

$$\begin{align*}
\cos(\Delta \gamma) &= \pm \sqrt{\frac{\cos(2\Delta \gamma) + 1}{2}} \\
\sin(\Delta \gamma) &= \frac{\sin(2\Delta \gamma)}{2\cos(\Delta \gamma)}
\end{align*}$$

There are two solutions calculated for $\Delta \gamma$ using equation (23), but only one solution set is correct. The following will explain how to choose the correct solution.

Selection of angles $\Delta \gamma_1$ and $\Delta \gamma_2$. This section will introduce the method for selecting the correct angle $\Delta \gamma$ by introducing the third-order moments.
The third-order moment equations are
\[
\begin{align*}
\mu_{30} &= A^3(A_1^3 \nu_{30} + A_1^3 \nu_{03} + 3A_1^2 A_2 \nu_{21} + 3A_1 A_3 \nu_{12}) + 3^o \nu_{n} \mu_{20} \\
&\quad - 3^o \nu_{n} \mu_{10} + \nu_{n} \mu_{02} \\
\mu_{03} &= A^3(A_1^3 \nu_{30} + A_2^3 \nu_{03} + 3A_1^2 A_2 \nu_{21} + 3A_1 A_3 \nu_{12}) + 3^o \nu_{n} \mu_{20} \\
&\quad - 3^o \nu_{n} \mu_{10} + \nu_{n} \mu_{02} \\
\mu_{12} &= A^3(A_1 A_2 A_3 \nu_{30} + (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12}) + 3^o \nu_{n} \mu_{10} + \nu_{n} \mu_{02} - \nu_{n} \mu_{01} \\
&\quad - 2^o \nu_{n} \nu_{o} \nu_{n} \mu_{01} + \nu_{n} \mu_{02} \\
\mu_{21} &= A^3(A_1 A_2 A_3 \nu_{30} + (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12}) + 3^o \nu_{n} \mu_{10} + \nu_{n} \mu_{02} - \nu_{n} \mu_{01} \\
&\quad - 2^o \nu_{n} \nu_{o} \nu_{n} \mu_{01} + \nu_{n} \mu_{02}
\end{align*}
\]

(24)

Substituting for $^o \nu_{n} \nu_{o} \nu_{n}$ using equation (12), we get
\[
\begin{align*}
\bar{\mu}_{30}(\Delta \gamma) &= A^3(K_{30}^a - K_{30}^b) - A^3(K_{30}^c - K_{30}^d) \\
\bar{\mu}_{03}(\Delta \gamma) &= A^3(K_{03}^a - K_{03}^b) - A^3(K_{03}^c - K_{03}^d) \\
\bar{\mu}_{12}(\Delta \gamma) &= A^3(K_{12}^a - K_{12}^b) - A^3(K_{12}^c - K_{12}^d) \\
\bar{\mu}_{21}(\Delta \gamma) &= A^3(K_{21}^a - K_{21}^b) - A^3(K_{21}^c - K_{21}^d)
\end{align*}
\]

(25)

where
\[
\begin{align*}
K_{30}^a &= A_1^3 \nu_{30} + A_2^3 \nu_{03} + 3A_1^2 A_2 \nu_{21} + 3A_1 A_3 \nu_{12} \\
K_{30}^b &= (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12} \\
K_{30}^c &= (A_1 A_2 A_3 \nu_{30} + (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12}) \\
K_{30}^d &= 2A_1^3 \nu_{03} - 3A_1 \mu_{02} \\
K_{03}^a &= A_1^3 \nu_{30} + A_2^3 \nu_{03} + 3A_1^2 A_2 \nu_{21} + 3A_1 A_3 \nu_{12} \\
K_{03}^b &= (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12} \\
K_{03}^c &= (A_1 A_2 A_3 \nu_{30} + (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12}) \\
K_{03}^d &= 2A_1^3 \nu_{03} - 3A_1 \mu_{02} \\
K_{12}^a &= A_1 A_2^2 \nu_{30} + (A_1 A_2 A_3 \nu_{30} + (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12}) \\
K_{12}^b &= (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12} \\
K_{12}^c &= (A_1 A_2 A_3 \nu_{30} + (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12}) \\
K_{12}^d &= 2A_1 A_2^2 \nu_{03} - 3A_1 \mu_{02} \\
K_{21}^a &= A_1 A_2^2 \nu_{30} + (A_1 A_2 A_3 \nu_{30} + (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12}) \\
K_{21}^b &= (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12} \\
K_{21}^c &= (A_1 A_2 A_3 \nu_{30} + (A_1 A_2^2 + 2A_1 A_4 A_2) \nu_{12}) \\
K_{21}^d &= 2A_1 A_2^2 \nu_{03} - 3A_1 \mu_{02}
\end{align*}
\]

From equation (19), the rotation matrix $^o \mathbf{R}_n$ can be calculated as
\[
^o \mathbf{R}_n = \begin{bmatrix}
\cos(\Delta \gamma) & -\sin(\Delta \gamma) & 0 \\
\sin(\Delta \gamma) & \cos(\Delta \gamma) & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(26)

Therefore, it is easy to know that $^o \mathbf{R}_n = \cos(\Delta \gamma)$, $^o \mathbf{R}_n = -\sin(\Delta \gamma)$, $^o \mathbf{R}_n = \sin(\Delta \gamma)$, and $^o \mathbf{R}_n = \cos(\Delta \gamma)$.

Finally, the relative rotation matrix $^o \mathbf{R}_n$ of the two camera frames $\mathbf{F}_n$ and $\mathbf{F}_o$ can be calculated by equation (13), where $^o \mathbf{R}_n$ and $^o \mathbf{I}_n$ are defined in equations (18), (19), and (14), respectively.

Estimate the translation vector of the two camera frames

This part will introduce the estimation method of the translation vector $^o \mathbf{t}_n$ in Figure 1(a).

From the camera frames $\mathbf{F}_n$ and $\mathbf{F}_o$ introduced in the previous section, we can obtain
\[
^o \mathbf{p} = ^o \mathbf{Z} \mathbf{R}_n \mathbf{t}_n \mathbf{p}
\]

(28)

where $^o \mathbf{R}_n$ and $^o \mathbf{R}_n$ are defined in equations (14) and (18), respectively, and $^o \mathbf{Z}$ and $^o \mathbf{Z}$ can be calculated as
\[
\begin{align*}
^o \mathbf{Z} &= \frac{1}{^o \mathbf{C}} \\
^o \mathbf{Z} &= \frac{1}{^o \mathbf{C}}
\end{align*}
\]

where $^o \mathbf{C}$ and $^o \mathbf{C}$ are expressed as
\[
\begin{align*}
^o \mathbf{C} &= ||^o \mathbf{R}_n^\top ^o \mathbf{N}|| = ||^o \mathbf{N}|| \\
^o \mathbf{C} &= ||^o \mathbf{R}_n^\top ^o \mathbf{N}|| = ||^o \mathbf{N}||
\end{align*}
\]
Substituting for \( \mathbf{p} \), \( \mathbf{p}' \), and \( \mathbf{R}_n \) using equations (28) and (13), equation (3) can be converted to

\[
\mathbf{o}' \mathbf{p} = \frac{n'}{o'Z} \mathbf{R}_n n' \mathbf{p} + \frac{1}{o'Z} \mathbf{R}_n \mathbf{o} t_n
\]  

(29)

Because the imaging planes of the camera frames \( \mathbf{F}_{n'} \) and \( \mathbf{F}_{o'} \) are parallel, \( n'Z \) and \( o'Z \) are constant. According to equations (29) and (12), the relationship between the image moments of cameras \( o' \) and \( n' \) can be expressed as

\[
\mathbf{o}' \mathbf{V} = \frac{n'}{o'Z} \mathbf{R}_n n' \mathbf{V} + \frac{1}{o'Z} \mathbf{R}_n \mathbf{o} \mathbf{t}_n
\]  

(30)

where \( \mathbf{o}' \mathbf{V} \) and \( n' \mathbf{V} \) are defined as

\[
\mathbf{o}' \mathbf{V} = \begin{bmatrix} \mu_{10} \\ \mu_{01} \\ 1 \end{bmatrix}, \quad n' \mathbf{V} = \begin{bmatrix} \nu_{10} \\ \nu_{01} \\ 1 \end{bmatrix}
\]

As a result, the translation vector \( \mathbf{t}_n \) is obtained

\[
\mathbf{o} \mathbf{t}_n = \mathbf{o}' \mathbf{R}_n ^T \left( \frac{n'}{o'Z} \mathbf{o}' \mathbf{V} - \frac{n'}{o'Z} \mathbf{R}_n n' \mathbf{V} \right)
\]  

(31)

Therefore, the calculation method of the homogeneous transformation matrix has been completed.

**Plane parameters estimation based on image moments**

The homogeneous transformation matrix calculation method introduced in the third section needs to know the plane parameters in the frames \( \mathbf{F}_o \) and \( \mathbf{F}_n \). In visual servo control, the latter is usually known, but the former is difficult to know. So we need to use the known information of image \( o \) and plane parameters \( \mathbf{N} \) to estimate plane parameters \( \mathbf{N} \). The methods introduced in the studies of Chaumette and Chaumette similarly apply only to typical primitives. Therefore, the following will present a plane parameters estimate method, which is suitable for any pattern.

Generally, the plane parameters \( \mathbf{N} \) are known, so we can calculate the rotation matrix \( \mathbf{R}_n \) and get the rotated image \( o' \). However, there are errors in the plane parameters \( \mathbf{N} \). Similarly, the rotation matrix \( \mathbf{R}_n \) and the rotated image \( n' \) can be calculated (note if there is no error in \( \mathbf{N}, n' = n' \)). Theoretically, the 2D rotation, 2D translation, and scale invariant moments of images \( o' \) and \( n' \) are the same. So we can correct the plane parameters \( \mathbf{N} \) according to these invariant moments.

We choose two moments \( \mathbf{\psi} = [\varphi_1, \varphi_2]^T \) that are invariant to 2D rotation, 2D translation, and scale^{25}

\[
\begin{align*}
\varphi_1 &= \frac{(c_{50} + 2c_{32} + c_{14})^2 + (c_{05} + 2c_{23} + c_{41})^2}{(c_{50} - 10c_{32} + 5c_{14})^2 + (c_{05} - 10c_{23} + 5c_{41})^2} \\
\varphi_2 &= \frac{(c_{50} - 2c_{32} - 3c_{14})^2 + (c_{05} - 2c_{23} - 3c_{41})^2}{(c_{50} - 10c_{32} + 5c_{14})^2 + (c_{05} - 10c_{23} + 5c_{41})^2}
\end{align*}
\]  

(32)

It is easy to know that \( n' \mathbf{\psi} = o' \mathbf{\psi} \), so we can calculate \( n' \mathbf{\psi} \) from image \( o' \). According to equation (9), the interaction matrix related to the invariant moments \( \mathbf{\psi} \) has the following form

\[
\begin{align*}
\mathbf{L}_{\psi_1}(A, B, c_{pq}) &= [\varphi_{1,1r}, \varphi_{1,1r}, \varphi_{1,1r}, \varphi_{1,1r}, 0] \\
\mathbf{L}_{\psi_2}(A, B, c_{pq}) &= [\varphi_{2,1r}, \varphi_{2,1r}, \varphi_{2,1r}, \varphi_{2,1r}, 0]
\end{align*}
\]  

(33)

We want to rotate image \( n' \) to image \( n' \) that is parallel to image \( o' \), so \( \mathbf{v}_e = 0 \) and \( \mathbf{\omega}_e \) will be calculated. Therefore, we can get

\[
\mathbf{\psi} = L_e \mathbf{\omega}_e
\]  

(34)

where

\[
L_e(c_{pq}) = \begin{bmatrix} \varphi_{1,1r} & \varphi_{1,1r} & 0 \\ \varphi_{2,1r} & \varphi_{2,1r} & 0 \end{bmatrix}
\]

(35)

Truncating the Taylor expansion at first order and substituting for \( \tilde{\mathbf{\psi}} \) using equations (34), we can approximate equation (35) as

\[
\tilde{\mathbf{\psi}} = \mathbf{\omega}_e \Delta t + O(\Delta t^2)
\]

(36)

Therefore, rotation matrix \( \mathbf{R}_{n'} \) of the frame \( \mathbf{F}_{n'} \) relative to \( \mathbf{F}_n \) can be calculated as

\[
\mathbf{n'} \mathbf{R}_{n'} = (\tilde{\mathbf{\psi}} \mathbf{R}_n) ^T = (\mathbf{\psi} \mathbf{R}_n) ^T
\]

(37)

where \( [\mathbf{\psi} \mathbf{R}_n]_e \) is a \( 3 \times 3 \) skew-symmetric matrix representation of \( \tilde{\mathbf{\psi}} \).

Now, we start to correct the initial plane parameters \( \mathbf{N} \). First, the image \( n' \) will be rotated to \( n' \) by the rotation matrix \( \mathbf{R}_{n'} \). Noting that

\[
\sqrt{S o'Z} \sqrt{o'm_{00}} = n'Z \sqrt{n'm_{00}}
\]

(38)

where \( S \) is the area of the object plane. Then, the plane parameter in the frame \( \mathbf{F}_{n'} \) is \( \mathbf{N} = [0 0 n']^T \). According to equations (29) and (38), \( n' \mathbf{C} \) can be expressed as

\[
n' \mathbf{C} = o' \mathbf{C} = \sqrt{o'm_{00}} \sqrt{n'm_{00}}
\]

(39)

The final corrected plane parameters \( \mathbf{N} \) are

\[
\mathbf{N} = (n' \mathbf{R}_{n'} \tilde{\mathbf{R}}_n) ^T n' \mathbf{N}
\]

(40)

Plane parameter error \( \mathbf{e}_N \) can be calculated as \( \mathbf{e}_N = \mathbf{N} - \mathbf{N} \), where the desired plane parameter \( \mathbf{N} \) can be approximately expressed as \( \mathbf{N} \approx \mathbf{N} \). Finally, we can design the correction scheme as

\[
\hat{\mathbf{e}}_N = \tilde{\mathbf{N}} - \mathbf{N}
\]

where \( \hat{\mathbf{e}}_N = \mathbf{N} - \mathbf{N} \).
PBVS and stability analysis

This section first introduces the PBVS control scheme, which is based on camera pose estimation described in the third section. Then, the stability of the method will be analyzed. This visual servoing scheme has been studied by Chaumette and Hutchinson.26 We briefly revisit the PBVS control scheme.

In this study, we can define the expected and current values of features $s^* = 0$ and $s = [\psi t_n \ \theta^\prime \omega_n]^T$, in which $\psi t_n$ is a translation vector calculated using equation (31) and $\theta^\prime \omega_n$ gives the angle/axis parameterization for the rotation matrix $\omega R_n$ calculated by equation (13). Therefore, the vision-based control schemes minimize an error $e(t)$, which is defined as

$$e(t) = s - s^* = [\psi t_n \ \theta^\prime \omega_n]^T$$

The relationship between $\dot{e}_s$ and $v_c$ is given by

$$\dot{e}_s = \dot{s} = L v_c$$

So the control scheme is obtained by

$$v_c = -\hat{L}_s^{-1} e_s$$

where $\hat{L}_s$ is an estimation of the interaction matrix $L_s$ related to $e_s$ and be calculated as

$$\hat{L}_s = \begin{bmatrix} 0 \ R_n \ 0 \\ 0 \ \ L_{\theta^\prime \omega_n} \end{bmatrix}$$

where $L_{\theta^\prime \omega_n}$ is given by

$$L_{\theta^\prime \omega_n} = I_3 - \frac{\theta}{2} [\omega_n]_x + \left(1 - \frac{\sin(\theta)}{\sin^2(\theta/2)}\right) [\omega_n]_x^2$$

where $\sin(x)$ is the sinus cardinal defined such that $x\sin(x) = \sin(x)$ and $\sin(0) = 1$. According to equation (43), rotation and translation are decoupled, which allows us to obtain a simple control scheme.

Figure 2. Results obtained using PBVS with known plane parameters: (a) “octopus” contour, (b) camera trajectory, (c) errors on pose, and (d) camera velocities. PBVS: position-based visual servoing.
Next, simply analyze the stability. Lyapunov function candidate is defined as

\[
V = \frac{1}{2} e_s^T e_s + \frac{1}{2} e_N^T e_N
\]  (45)

By deriving the Lyapunov function concerning time, equation (45) was transformed into

\[
\dot{V} = e_s^T \dot{e}_s + e_N^T \dot{e}_N
\]

\[
= -\lambda_s e_s^T L_s^{-1} e_s - \lambda_N e_N^T e_N
\]  (46)

If \( ^wN = ^wN \) is satisfied and the object plane is always in the camera field of view, then both

\[
-\lambda_s e_s^T L_s^{-1} e_s \approx -\lambda_s e_s^T e_s < 0
\]
and

\[-\hat{\lambda} e_N^T e_N \approx -\hat{\lambda} e_N^T e_N < 0\]

will be guaranteed. So \(\dot{V} < 0\), the system is asymptotically stable.

**Simulation results**

We evaluate the control scheme proposed in this article. Considering a vision sensor and an object plane as examples, the co-simulation was conducted on the MATLAB 2020b and CoppeliaSim 4.1 platforms. Some different patterns, that is, an “octopus,” a “whale,” and a “flame” (see Figures 2(a), 4(a), and 5(a)), were considered in the plane. The white noise with standard deviation \(\sigma = 0.1\) pixels is added to the image in the following simulations.

**Simulation results using camera pose estimation**

This part mainly verifies the effectiveness of the camera pose estimation method introduced in the third section, so we assume that plane parameters in the frame \(F_n\) are known (\(N = [-0.3716 - 0.0567 0.5307]^T\)). We consider the case where the image and object planes with the “octopus” pattern are nonparallel at the desired position (\(\hat{N} = [0.2986 0.1526 0.8273]^T\)). Figure 2(a) shows the...
Figure 5. Results obtained using two methods with known plane parameters: (a) “frame” contour, (b) camera trajectory, (c) errors on features using classical method, (d) errors on features using our method, (e) errors on pose using classical method, (f) errors on pose using our method, (g) camera velocities using classical method, and (h) camera velocities using our method.
green and red contours representing the desired and initial image contours, respectively. Then, PBVS scheme is used for camera control.

The obtained results are given in Figure 2. They show the good behavior of camera pose estimation and the control law. The translation component of the visual features is \( \mathbf{u}_b = \begin{bmatrix} x_b^t & y_b^t & z_b^t \end{bmatrix}^T \), so the obtained camera trajectory is straight line (see Figure 2(b)). Despite the corresponding displacement is very large (\( t_z = 0.1084 \text{ m}, \ t_y = 0.3403 \text{ m}, \ t_x = -0.5524 \text{ m}, \ \theta_{\omega_x} = 0.1370 \text{ rad}, \ \theta_{\omega_y} = 0.4581 \text{ rad}, \ \text{and} \ \theta_{\omega_z} = -2.1889 \text{ rad} \)), we can note in Figure 2(c) and (d) the decoupled and exponential decrease of the six combinations of the corresponding displacement and the six camera velocity components. Because the interaction matrix is decoupled, the corresponding displacement (see Figure 2(c)) all converge to less than \( 10^{-4} \). Although noise is added, the pose estimation method proposed in this article is suitable when the image and object planes are nonparallel at the desired position.

In addition, the method proposed in this article is suitable for the case that \( |\theta_{\omega_z}| > \pi/2 \), but some methods based on image moments, that is, the methods proposed by Chaumette\(^1\) and Tahri and Chaumette,\(^2\) do not have this advantage. The following will design two simple situations to illustrate this problem.

The visual features in the method of Tahri and Chaumette\(^2\) are

\[
\mathbf{s} = [x_n \ y_n \ a_n \ \varphi_1 \ \varphi_2 \ \alpha]^T
\]

(47)

where \( \varphi_1 \) and \( \varphi_2 \) can be calculated by equation (32), and

\[
\begin{align*}
x_n &= a_n x_g \\
y_n &= a_n y_g \\
a_n &= o Z \sqrt{\frac{m_{00}}{m_{00}}}
\end{align*}
\]

\[
\alpha = \frac{1}{2} \arctan \left( \frac{2c_{11}}{c_{20} - c_{02}} \right)
\]

The “octopus” contour continues to be adopted, but \( \theta_{\omega_x} = 0, \ \theta_{\omega_y} = 0 \) (equivalent to \( A = 0, B = 0 \)). One situation is \( |\theta_{\omega_z}| < \pi/2 \), the other is \( |\theta_{\omega_z}| > \pi/2 \) (see Figure 3(a) and (b)). The obtained results are given in Figure 3. The method in this article can make the feature errors \( \mathbf{e}_x \) and pose errors \( [(e_{x1}^x \ e_{y1}^x \ e_{z1}^x \ e_{\theta_{\omega_x}}^x \ e_{\theta_{\omega_y}}^x \ e_{\theta_{\omega_z}}^x)]^T \) converge to less than \( 10^{-4} \) for both situations. However, the method proposed by Tahri and Chaumette\(^2\) is only suitable for case that \( |\theta_{\omega_z}| < \pi/2 \) (see Figure 3(c) and (e)). For \( |\theta_{\omega_z}| > \pi/2 \), this method can only converge feature error \( \mathbf{e}_x \) to less than \( 10^{-4} \) (see Figure 3(d)), while the pose error \( [(e_{x1}^x \ e_{y1}^x \ e_{z1}^x \ e_{\theta_{\omega_x}}^x \ e_{\theta_{\omega_y}}^x \ e_{\theta_{\omega_z}}^x)]^T \) still has a large error, that is, \( e_{\theta_{\omega_z}} = -\pi \) (see Figure 3(f)). The reason is that the feature of the velocity \( \omega_z \) is \( \alpha \), which is only available for \( |\theta_{\omega_z}| \in [-\pi/2, \pi/2] \). Therefore, the calculation of \( \bar{\mathbf{R}}_p \), proposed in “The calculation of the rotation matrix \( \bar{\mathbf{R}}_p \)” section has certain advantages for case that \( |\theta_{\omega_z}| > \pi/2 \).

**Simulation results using plane parameters estimation**

This part mainly verifies the plane parameters estimation method introduced in the fourth section. We simulated a situation where a desired image and a object plane with the “whale” pattern are nonparallel at the desired position. \( \mathbf{N} = [0.6016 \ 0.2122 \ 1.0419]^T \) and \( ^a\mathbf{N} = [-0.3716 - 0.0567 \ 0.5307]^T \). Twenty-seven percent errors are randomly added to \( ^a\mathbf{N} \) to obtain plane parameters \( ^a\mathbf{N} = [-0.4617 - 0.0775 0.6778]^T \). Note that \( ^a\mathbf{N} \) is unknown, but \( ^a\mathbf{N} \) is known. Figure 4(a) shows the green and red contours representing the desired and initial image contours, respectively. The PBVS schemes with and without plane parameters estimation are used to control the cameras, respectively.

The obtained results are given in Figure 4. The camera trajectories controlled by the PBVS schemes with and without plane parameters estimation are shown in Figure 4(b). Due to the early correction of plane parameter errors (Figure 4(c)), the trajectory obtained by the former method is slightly different from the straight-line trajectory in
Figure 7. Results obtained using two methods with errored plane parameters: (a) camera trajectory, (b) errors on plane parameters, (c) errors on features using classical method, (d) errors on features using our method, (e) errors on pose using classical method, (f) errors on pose using our method, (g) camera velocities using classical method, and (h) camera velocities using our method.
Figure 2(b). It is easy to see that the plane parameter errors of the former method \((\mathbf{e}_A^s, \mathbf{e}_B^s, \mathbf{e}_C^s)^T\) quickly converge to \(10^{-4}\), while the plane parameter errors of the latter method \((\mathbf{ar{e}}_A, \mathbf{ar{e}}_B, \mathbf{ar{e}}_C)^T\) still have a large error in Figure 4(c). Despite the corresponding displacement is large \((t_x = -0.9133 \text{ m}, t_y = -0.7553 \text{ m}, t_z = -0.1625 \text{ m}, \theta \omega_x = -0.2980 \text{ rad}, \theta \omega_y = 1.0479 \text{ rad}, \text{ and } \theta \omega_z = 1.1873 \text{ rad})\), the pose errors obtained by the former method still all converge to less than \(10^{-4}\), but the pose errors \((\mathbf{e}_A^3, \mathbf{e}_B^3, \mathbf{e}_C^3, \mathbf{e}_A^4, \mathbf{e}_B^4)^T\) obtained by the latter cannot converge to a small value (see Figure 4(d)). The PBVS scheme shows camera velocities (see Figure 4(e) and (f)). There is no oscillation for any component of the camera velocity in both methods.

As a result, the plane parameters estimation method proposed in this article can effectively eliminate the parameter error, and satisfactory results are obtained in PBVS scheme.

**Simulation results compared to the classical method**

This part will show the comparison results between the method proposed in this article and the classical method proposed by Tahri and Chaumette.\(^{12}\) The latter’s visual features are expressed as equation (47) and the interaction matrix is

\[
\mathbf{L}_x = \frac{1}{2}(\mathbf{L}_x + \mathbf{L}_y)
\]

which largely improves the system behavior\(^{28}\) The former’s interaction matrix is calculated by equation (43). We consider the case where the image and object planes with the “frame” pattern (see Figure 5(a)) are parallel at the desired position \((\mathbf{^wN} = [0 \ 0 \ 1.1765]^T \text{ and } \mathbf{^wN} = [-0.0083 \\ -0.2941 \ 0.5290]^T)\). The corresponding displacement is \((t_x = -1.0735 \text{ m}, t_y = -0.4274 \text{ m}, t_z = -0.8021 \text{ m}, \theta \omega_x = -0.4925 \text{ rad}, \theta \omega_y = 0.1439 \text{ rad}, \text{ and } \theta \omega_z = -0.5007 \text{ rad})\).

Firstly, we assume that the planar parameter \(^wN\) is known, and then use these two methods for visual servo control, respectively. The obtained results are shown in Figure 5.

The camera trajectories controlled by the two methods are showed in Figure 5(b). Because our method adopts PBVS control scheme, the camera trajectory is a straight line; while the classical method adopts the IBVS control scheme, so the camera trajectory is a curve. The two methods can not only make the feature errors converge to \(10^{-4}\) (see Figure 5(c) and (d)) but also make the pose errors converge to \(10^{-4}\) (see Figure 5(e) and (f)). However, the camera velocities controlled by our method is more stable than the camera velocities controlled by the classical method (see Figure 5(g) and (h)). The result is that the interaction matrix computed by our method has a better condition number than the classical method. Because the former has a completely decoupled interaction matrix (equation (43)). The boxplots of condition numbers of interaction matrices obtained by two methods with errored plane parameters are shown in Figure 6. The maximum, minimum, and mean values of the condition numbers calculated by the classical method are 3206.2393, 65.1757, and 209.3060, respectively. However, the maximum, minimum, and mean values of the condition numbers calculated by the our method are 1.0217, 1.000000, and 1.0014, respectively. Therefore, the method proposed in this article has obvious advantages over the classical method when the planar parameter \(^wN\) is known.

Finally, 30% errors are randomly added to \(^wN\) to obtain planar parameters \(^wN = [-0.0116 - 0.1860 \ 0.3650]^T\). Then, the two methods are respectively used for visual servo control. The obtained results are shown in Figure 7.

The camera trajectories controlled by the two methods are shown in Figure 7(a). Although there are errors in plane parameters, both methods can converge the visual features to \(10^{-4}\) (see Figure 7(c) and (d)). However, our method can also converge the pose error to \(10^{-4}\), while the
classical method can only converge to $10^{-2}$ (see Figure 7(e) and (f)). This is because the former method has plane parameter estimation and quickly converges the plane parameter errors to $10^{-4}$, but the latter method does not have this advantage (see Figure 7(b)). It can be seen from Figure 7(g) and (h) that the camera velocities calculated by the method proposed in this article are still more stable than the camera velocities calculated by the classical method. The boxplots of condition numbers of interaction matrices obtained by the two methods with errored plane parameters are shown in Figure 8. The maximum, minimum, and mean values of the condition numbers calculated by the classical method are 2244.6194, 65.6998, and 130.4685, respectively. However, the maximum, minimum, and mean values of the condition numbers calculated by our method are 1.0214, 1.0000, and 1.0015, respectively. The latter is significantly smaller than the former. Therefore, the method proposed in this article still has obvious advantages over the classical method when the planar parameter $^bN$ is errored.

Conclusion
This study proposes two new estimation methods based on image moments, which are used to estimate camera pose and plane parameters, respectively. The former method, which is a closed-form solution, directly estimates the relative pose of the initial camera and the desired camera, not the pose of the camera relative to the object plane. The latter method uses the 2D rotation, 2D translation, and scale invariant moments to estimate the plane parameters. From both estimation methods, the article employs a PBVS scheme for the object plane. The simulation results have validated our approaches. One advantage of the two estimation methods is that they do not require image processing such as features extraction, matching, and tracking, the other is that the PBVS scheme is suitable for the case that $|\theta_{\omega_j}| > \pi/2$. In addition, the condition number of the interaction matrix calculated by the method proposed in this article is very small, which is very important for visual servo control. However, some particular configurations can make the object plane leave the camera field of view, which will affect the stability of visual servoing control. Future work will be devoted to designing a visual servoing scheme that increases the stability and prefer better decoupling of the interaction matrix.

Acknowledgment
The authors thank Shuo Wang for helping with the language of this manuscript and all colleagues for providing different types of help during the preparation of this manuscript.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by National Key R&D Program of China (2016YFC0803000, 2016YFC0803005).

ORCID iDs
Yuhan Chen https://orcid.org/0000-0001-9796-0291
Xiao Luo https://orcid.org/0000-0003-2574-4594

References
1. Fathian K, Jin J, Wee SG, et al. Camera relative pose estimation for visual servoing using quaternions. Robot Autonom Syst 2018; 107: 45–62.
2. Bakthavatchalam M, Tahri O, and Chaumette F. Improving moments-based visual servoing with tunable visual features. In: 2014 IEEE international conference on robotics and automation (ICRA), 31 May–5 June 2014, Hong Kong, China, pp. 6186–6191. IEEE.
3. Espiau B and Chaumette F. A new approach to visual servoing in robotics. IEEE Transact Robot & Automat 1992; 8(3): 313–326.
4. Li CL, Cheng MY, and Chang WC. Dynamic performance improvement of direct image-based visual servoing in contour following. Int J Adv Robot Syst 2018; 15(1): 1729881417753859.
5. Bueno-López M and Arteaga-Pérez MA. Fuzzy vs nonfuzzy in 2D visual servoing for robot manipulators. Int J Adv Robot Syst 2013; 10(2): 108.
6. Colombo FT, de Carvalho Fontes JV, and da Silva MM. A visual servoing strategy under limited frame rates for planar parallel kinematic machines. J Intelligent Robot Syst 2019; 96(1): 95–107.
7. Noh S, Park C, and Park J. Position-based visual servoing of multiple robotic manipulators: verification in gazebo simulator. In: 2020 international conference on information and communication technology convergence (ICTC), 21–23 October 2020, Jeju Island, South Korea, pp. 843–846. IEEE.
8. Xin J, Chen K, Bai L, et al. Depth adaptive zooming visual servoing for a robot with a zooming camera. Int J Adv Robot Syst 2013; 10(2): 120.
9. Bakthavatchalam M, Tahri O, and Chaumette F. A direct dense visual servoing approach using photometric moments. IEEE Transact Robot 2018; 34(5): 1226–1239.
10. Chaumette F and Hutchinson S. Visual servoing and visual tracking. In: Siciliano B, Khatib O, and Ang MH (eds) Handbook of robotics. Berlin: Springer, 2008, pp. 563–583.
11. Chaumette F. Image moments: a general and useful set of features for visual servoing. IEEE Transact Robot 2004; 20(4): 713–723.
12. Tahri O and Chaumette F. Point-based and region-based image moments for visual servoing of planar objects. IEEE Transact Robot 2005; 21(6): 1116–1127.
13. Tahri O, Tamtsia AY, Mezouar Y, et al. Visual servoing based on shifted moments. IEEE Transact Robot 2015; 31(3): 798–804.
Appendix I

The relationship between the rotation matrix \( {}^o\mathbf{R}_n \) and
\( Z - Y - X \) Euler angles

This section will introduce the relationship between the rotation matrix \( {}^o\mathbf{R}_n \) and \( Z - Y - X \) Euler angles.

To calculate the rotation matrix \( {}^o\mathbf{R}_n \) in Figure 1(a), it is
assumed that there is a frame \( w' \), then the rotation matrix
\( {}^o\mathbf{R}_n \) can be calculated as

\[
{}^o\mathbf{R}_n = (w'\mathbf{R}_o)^T w\mathbf{R}_n \tag{49}
\]

The two rotation matrices \( w'\mathbf{R}_o \) and \( w'\mathbf{R}_n \) are expressed
using \( Z - Y - X \) Euler angles as

\[
\begin{align*}
{}^o\mathbf{R}_o &= \mathbf{R}_z(\alpha_0) \cdot \mathbf{R}_y(\beta_0) \cdot \mathbf{R}_x(\alpha_0) \\
{}^o\mathbf{R}_n &= \mathbf{R}_z(\gamma_0) \cdot \mathbf{R}_y(\beta_0) \cdot \mathbf{R}_x(\alpha_0) \tag{50}
\end{align*}
\]

Injecting equation (50) into equation (49), \( {}^o\mathbf{R}_n \) is written as

\[
{}^o\mathbf{R}_n = \mathbf{R}_z^{T}(\alpha_0) \mathbf{R}_y^{T}(\beta_0) \mathbf{R}_x^{T}(\gamma_0) \mathbf{R}_z(\gamma_0) \cdot \mathbf{R}_y(\beta_0) \cdot \mathbf{R}_x(\alpha_0) \\
= \left( \mathbf{R}_z(\alpha_0) \mathbf{R}_x(\alpha_0) \right)^{T} \mathbf{R}_z(\gamma_0) \mathbf{R}_y(\beta_0) \mathbf{R}_x(\alpha_0) \tag{51}
\]

By defining \( n'\mathbf{R}_n = \mathbf{R}_z(0)\mathbf{R}_y(\beta_0)\mathbf{R}_x(\alpha_0) \), \( o'\mathbf{R}_n = \mathbf{R}_z(0)\mathbf{R}_y(\beta_0)\mathbf{R}_x(\alpha_0) \), and \( o'\mathbf{R}_n = \mathbf{R}_z(\gamma_0)\mathbf{R}_y(\beta_0)\mathbf{R}_x(\alpha_0) \), equation (51) can
be expressed as \( o'\mathbf{R}_n = o'\mathbf{R}_n - o'\mathbf{R}_n \). Note
that angles \( o_\gamma \) and \( o_\gamma \) are the rotations of the camera frame
around the z-axis, and angles \( o_\gamma \) and \( o_\gamma \) in equation (19)
are the rotations of the camera images around the z-axis, that is,
\( \mathbf{R}_z(\gamma_0 - o_\gamma) = \mathbf{R}_z(o_\gamma - \gamma_0) \).