A new pose detection method for self-calibrated cameras based on parallel lines and its application in visual control system

De Xu[1,2], You Fu Li[1*], Yang Shen[2], Min Tan[2]

[1]Department of Manufacturing Engineering and Engineering Management,

City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong

Email: sdxude@yahoo.com, meyfli@cityu.edu.hk

[2] The Key Laboratory of Complex System and Intelligence Science,

Institute of Automation, Chinese Academy of Sciences, Beijing 100080, P. R. China

Email: yang.shen@mail.ia.ac.cn, tan@compsys.ia.ac.cn

(*Corresponding author)

Abstract: In this paper, a new method is proposed to detect the pose of an object with two cameras. Firstly, the intrinsic parameters of the cameras are self-calibrated with two pairs of parallel lines, which are orthogonal. Then the poses of the cameras relative to the parallel lines are deduced, and the rotational transformation between the two cameras is calculated. With the intrinsic parameters and the relative pose of the two cameras, a method is proposed to obtain the poses of a line, plane and rigid object. Furthermore, a new visual control method is developed using pose detection rather than 3D reconstruction. Experiments are conducted to verify the effectiveness of the proposed method. The method can also be applied to mobile manipulators, humanoid robots and so on.

Key words: Pose detection, camera self-calibration, ratio invariance, hand-eye system, visual control.

1 Introduction

Detecting the pose of an object is needed in many robotic tasks such as tracking and grasping. Pose detection has been used in Position Based Visual Servoing (PBVS) and Hybrid Visual Servoing (HVS) [1]. For example, Han *et al* [2] provided a method for door opening with a stereovision system using some special marks. The marks consisted of some rectangles where the cross points between the main rectangle and the small ones were selected as feature points. The poses of the doorknob and the end-effector were calculated from the positions, measured with stereovision, of the feature points. Malis *et al* [3] proposed a 2.5D visual servoing method that dealt with position control in image space using extended image coordinates, and pose control in Cartesian space according to the pose obtained through 3D reconstruction. Hespanha *et al* [4] investigated the performances of three visual servoing methods, PBVS, HVS and Image Based Visual Servoing (IBVS) with un-calibrated cameras. Although IBVS performed best in position tracking, it is hard to be used for pose control. Therefore, pose detection is important for tasks requiring pose adjustments and control.

Pose detection methods can be classified into different categories as follows.

(1) Using known knowledge of an object. For example, Kragic *et al* [5] estimated the pose and position of an object according to its CAD model and images. Sugimoto *et al* [6] gave a method to estimate the translation and rotation of a mobile robot using line and intersection constraints at two views in indoor scenes. The parameters of the stereovision system were pre-calibrated. Yang *et al* [7]

developed a method to measure the pose of symmetric objects, and gave application examples in robot navigation.

(2) Via 3D reconstruction. For example, Sato *et al* [8] proposed a projection reconstruction method based on robot's movement offset and epipolar geometry at two views. They calculated the direction of the object relative to the camera mounted on a mobile robot. Pose detection using 3D reconstruction can be found in many literatures [2, 3]. With this method, the errors in positioning will be introduced into the pose calculation.

(3) Via estimation of image Jacobian matrix. This method is widely used in un-calibrated visual servoing [9-15]. However, singularity in the image Jacobian matrix presents stability problem in the pose estimation. Schramm *et al* [15] presented a method to improve the stability through estimating the camera parameters and the robot Jacobian matrix separately.

For applications in an indoor environment, the first type of the above methods is more promising because of the available environment knowledge. Although the objects may have limited features to be used for camera calibration, the environment probably provides more. Therefore, the camera's self-calibration can be achieved using constraints in an environment. 3D reconstruction suffers from correspondence problem especially when point matching is used. In this regard, line matching would offer a better solution to pose detection than point matching. Yet, direct 3D reconstruction should be avoided in pose detection to reduce the influence of positioning errors.

The pose of a line in Cartesian space cannot be determined with one camera at single view. To solve this problem, one can either use multiple views for a single camera or use two cameras instead. In either case, the intrinsic parameters of the cameras need to be calibrated. In man-made environments, there often exist parallel lines that can be employed to calibrate cameras. In this regard, Carvalho *et al* [16] presented a case of camera calibration with the constraints existing in football field. As known, the vanishing point contains the information of the intrinsic parameters. Many references about camera calibration of intrinsic parameters with vanishing points can be found [17-19]. Bénallal *et al* [17] proposed a method to calibrate a camera with three orthogonal vanishing points. Generally, it is not easy to find three orthogonal vanishing points in an indoor environment. Guillou *et al* [18] developed a method using two orthogonal vanishing points in a single image. However, the principal point of the camera was supposed at the center of the image. Sometimes, this assumption may not be valid. Lee *et al* [19] gave a method using two orthogonal vanishing points from image streams without the assumption of known principal point. The candidate space of principal point and focal length were derived from the relation of multiple hemispheres. In [17-19], the intrinsic parameters of a camera consisted of the focus length and principal point. The difference between the horizontal and vertical magnification was not considered. In addition, it is not easy to find the vanishing points with satisfactory accuracy in a natural image [20, 21].

In this paper, the self-calibration of a pair of cameras, including the intrinsic parameters and the transformation between the cameras, is achieved using parallel line constraint. A new method to detect the pose of an object with two cameras is proposed. Throughout this paper, the term pose refers to orientation only. A new visual control method is presented using pose detection rather than 3D reconstruction. Our method is easier to implement than a traditional method like a position-based one. With our method, the tedious calibration of the extrinsic parameters of cameras is avoided and the position information of the environment or metric information of the object is no longer needed. The rest of the paper is organized as follows. The camera model is introduced in Section 2. Section 3 investigates the self-calibration of the intrinsic parameters using parallel lines. In Section 4, a new method for calibrating the transformation between two cameras is developed based on orthogonal parallels. The pose detection methods for a line, plane and rigid object are also presented. Section 5 proposes a new hybrid visual control scheme using the poses of the object and end-effector without 3D reconstruction. Sensitivity analyses are presented in Section 6. The experimental results are given in Section 7. Finally, we conclude the paper in Section 8.

2 Camera model

In this paper, we assume a pinhole camera model with lens distortion negligible. If the lens distortion is noticeable, it can be corrected in advance by a method we developed in a separate work [24]. The four parameter model of a camera can be given as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c / z_c \\ y_c / z_c \\ 1 \end{bmatrix} = M_{in} \begin{bmatrix} x_c / z_c \\ y_c / z_c \\ 1 \end{bmatrix} \tag{1}$$

where [u, v] are the coordinates of a point in an image, [u_0, v_0] denote the image coordinates of the camera's principal point, [x_c, y_c, z_c] are the coordinates of a point in the camera frame, M_{in} is the intrinsic parameters matrix, and [k_x, k_y] are the magnification coefficients from the imaging plane coordinates to the image coordinates.

Assume that the camera frame is denoted as $\mathbb{C}$, and the work space frame as $\mathcal{W}$. The transformation from $\mathbb{C}$ to $\mathcal{W}$ is known as the extrinsic parameters for the camera

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = {}^{c}M_w \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2}$$

where [x_w, y_w, z_w] are the coordinates of a point in the object frame, and ${}^{c}M_w$ is the extrinsic parameter matrix of the camera, i.e. the transformation from the frame $\mathbb{C}$ to $\mathcal{W}$. In ${}^{c}M_w$, $\vec{n} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T$ is the direction vector of the x-axis, $\vec{o} = \begin{bmatrix} o_x & o_y & o_z \end{bmatrix}^T$ is that of the y-axis, $\vec{a} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$ is that of the z-axis for the frame $\mathcal{W}$ expressed in the frame $\mathbb{C}$, and $\vec{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$ is the position vector.

3 Self-calibration of the intrinsic parameters

Here, a self-calibration method for the intrinsic model with four parameters is studied using two planar groups of orthogonal parallels, which are popular in the indoor environment.

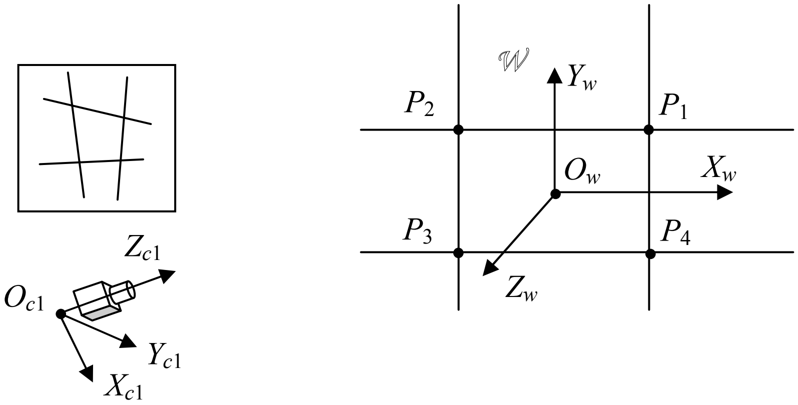


Fig. 1 Parallels and their imaging lines

As shown in Fig.1, a rectangle is formed with two groups of orthogonal parallels, which is projected as a quadrangle on the image plane of a camera. Assume that the four vertexes P_1 to P_4 have the coordinates as [a, b, 0], [-a, b, 0], [-a, -b, 0] and [a, -b, 0] in a Cartesian frame established at the quadrangle's center, and [u_1, v_1], [u_2, v_2], [u_3, v_3], [u_4, v_4] in image space. Combining equation (1) and (2), we have [22]

$$z_c\begin{bmatrix}u\\v\\1\end{bmatrix}=M_{in}{}^{w}M_c\begin{bmatrix}x_w\\y_w\\z_w\\1\end{bmatrix}=\begin{bmatrix}m_{11}&m_{12}&m_{13}&m_{14}\\m_{21}&m_{22}&m_{23}&m_{24}\\m_{31}&m_{32}&m_{33}&m_{34}\end{bmatrix}\begin{bmatrix}x_w\\y_w\\z_w\\1\end{bmatrix}\tag{3}$$

Applying the coordinates of the four points to equation (3), and eliminating the unknown parameter z_c, we obtain

$$u_1m_{31}a+u_1m_{32}b+u_1m_{34}=m_{11}a+m_{12}b+m_{14}\tag{4}$$

$$v_1m_{31}a+v_1m_{32}b+v_1m_{34}=m_{21}a+m_{22}b+m_{24}\tag{5}$$

$$-u_2m_{31}a+u_2m_{32}b+u_2m_{34}=-m_{11}a+m_{12}b+m_{14}\tag{6}$$

$$-v_2m_{31}a+v_2m_{32}b+v_2m_{34}=-m_{21}a+m_{22}b+m_{24}\tag{7}$$

$$-u_3m_{31}a-u_3m_{32}b+u_3m_{34}=-m_{11}a-m_{12}b+m_{14} \tag{8}$$

$$-v_3m_{31}a-v_3m_{32}b+v_3m_{34}=-m_{21}a-m_{22}b+m_{24} \tag{9}$$

$$u_4m_{31}a-u_4m_{32}b+u_4m_{34}=m_{11}a-m_{12}b+m_{14} \tag{10}$$

$$v_4m_{31}a-v_4m_{32}b+v_4m_{34}=m_{21}a-m_{22}b+m_{24} \tag{11}$$

Combining equations (4), (6), (8), (10), and equations (5), (7), (9), (11) respectively, we can derive

$$\begin{cases}(u_1-u_3-u_4+u_2)m'_{31}+(u_1-u_3-u_2+u_4)m'_{32}=u_2+u_4-u_1-u_3\\(v_1-v_3-v_4+v_2)m'_{31}+(v_1-v_3-v_2+v_4)m'_{32}=v_2+v_4-v_1-v_3\end{cases} \tag{12}$$

here $m'_{31}=m_{31}a/m_{34}$, $m'_{32}=m_{32}b/m_{34}$, $m_{34}=p_z>0$.

The variables m'_{31} and m'_{32} can be resolved with equation (12). Define the temporary variables $m'_{11}=m_{11}a/m_{34}$, $m'_{12}=m_{12}b/m_{34}$, $m'_{14}=m_{14}/m_{34}$, $m'_{21}=m_{21}a/m_{34}$, $m'_{22}=m_{22}b/m_{34}$ and $m'_{24}=m_{24}/m_{34}$. They can be computed with Least Square Method via substituting m'_{31} and m'_{32} into equations (4) to (11). Then the following can be deduced according to equation (3).

$$\begin{cases}n_x=\dfrac{(m'_{11}/m'_{31}-u_0)n_z}{k_x},\ n_y=\dfrac{(m'_{21}/m'_{31}-v_0)n_z}{k_y}\\o_x=\dfrac{(m'_{12}/m'_{32}-u_0)o_z}{k_x},\ o_y=\dfrac{(m'_{22}/m'_{32}-v_0)o_z}{k_y}\end{cases} \tag{13}$$

Notice that vector $\vec{n}$ is orthogonal to vector $\vec{o}$. Thus

$$\left[\frac{(m'_{11}/m'_{31}-u_0)(m'_{12}/m'_{32}-u_0)}{k_xk_x}+\frac{(m'_{21}/m'_{31}-v_0)(m'_{22}/m'_{32}-v_0)}{k_yk_y}+1\right]n_zo_z=0 \tag{14}$$

If $n_z\neq0$ and $o_z\neq0$, then

$$\frac{(m'_{11}/m'_{31}-u_0)(m'_{12}/m'_{32}-u_0)}{k_x k_x}+\frac{(m'_{21}/m'_{31}-v_0)(m'_{22}/m'_{32}-v_0)}{k_y k_y}+1=0 \tag{15}$$

Changing the camera's pose relative to the parallel lines, many nonlinear equations as given in (15) can be generated. With the temporary variables defined in (16), equation (17) can be deduced from two equations as shown in (15) [23].

$$\begin{cases} h_1 = k_x^2 \\ h_2 = k_x^2 / k_y^2 \\ h_3 = v_0 h_2 \end{cases} \tag{16}$$

$$(u_{i1}+u_{i2}-u_{j1}-u_{j2})u_0+(v_{i1}+v_{i2}-v_{j1}-v_{j2})h_3-(v_{i1}v_{i2}-v_{j1}v_{j2})h_2=u_{i1}u_{i2}-u_{j1}u_{j2} \tag{17}$$

where $[u_{i1}, v_{i1}]=[m'_{i11}/m'_{i31}, m'_{i21}/ m'_{i31}]$, $[u_{i2}, v_{i2}]=[m'_{i12}/m'_{i32}, m'_{i22}/m'_{i32}]$, $[u_{j1}, v_{j1}]= [m'_{j11}/m'_{j31}, m'_{j21}/ m'_{j31}]$, $[u_{j2}, v_{j2}]=[m'_{j12}/m'_{j32}, m'_{j22}/m'_{j32}]$.

There are three unknown parameters in equation (17), u_0, h_2 and h_3. They can be resolved with three or more equations. Then v_0 is calculated with h_2 and h_3 according to the third formula in (16). Submitting parameters u_0, v_0 and h_2 into equation (15), we have parameter k_x. Finally, k_y is computed from the second formula in (16) with k_x and h_2. It can be found that at least four equations as shown in (15) are needed for the resolving of the intrinsic parameters such as u_0, v_0, k_x and k_y. In other words, at least four views of the two groups of orthogonal parallels are necessary to determine the four intrinsic parameters.

Discussions:

(1) If n_z=0, the optical axis of the camera is perpendicular to the two parallel horizontal sides of the

rectangle. Therefore, the horizontal sides are also parallel lines in the image space. In other words, there is no intersection between the horizontal sides in the image space. In this case, equation (15) is not satisfied because of the absence of the vanishing point between the horizontal sides. Similarly, if o_z=0, there does not exist a vanishing point in the image space between the perpendicular sides. If n_z=0 and o_z=0, the optical axis of the camera is perpendicular to the rectangle. There will be no vanishing point in the image space. Therefore, it is a necessary condition for the intrinsic parameter calibration based on parallels that the camera's optical axis must not be perpendicular to the parallels.

To prevent equation (15) from being ill-conditioned, the parallelism of the parallels in the image space should be checked. The cosine function of the angle between two lines in the image space can be employed as the parallelism index, as given below

$$F_p = \cos\angle(L_i, L_j) = \frac{\left|(u_{i1}-u_{i2})(u_{j1}-u_{j2})+(v_{i1}-v_{i2})(v_{j1}-v_{j2})\right|}{\sqrt{(u_{i1}-u_{i2})^2+(v_{i1}-v_{i2})^2}\sqrt{(u_{j1}-u_{j2})^2+(v_{j1}-v_{j2})^2}} \tag{18}$$

where $[u_{i1}, v_{i1}]$ and $[u_{i2}, v_{i2}]$ are the image coordinates of two points on a line L_i, $[u_{j1}, v_{j1}]$ and $[u_{j2}, v_{j2}]$ are the image coordinates of another two points on a line L_j, $\angle(L_i, L_j)$ is the angle between the two lines L_i and L_j in the image space, F_p is the parallelism index between L_i and L_j.

If F_p=1, the imaging lines of the parallels are parallel. The camera's optical axis is perpendicular to the parallels. If $|F_p$-1$|<\varepsilon$, where ε is an infinitely small positive value, then the imaging lines of the parallels are almost parallel. In this case, equation (15) is also ill-conditioned. The smaller F_p is, the more robust equation (15) will be. Therefore, in self-calibrating the camera's intrinsic parameters, equation (18) can be used to check if a pair of parallels is acceptable or not to form an equation as given in (15).

(2) If $[u_0, v_0]$ is known in advance, assuming $k_x=k_y=k$, then k can be obtained with one view of two groups of parallels, as given below

$$k = \sqrt{-(m'_{11}/m'_{31} - u_0)(m'_{12}/m'_{32} - u_0) - (m'_{21}/m'_{31} - v_0)(m'_{22}/m'_{32} - v_0)} \tag{19}$$

here, k is a parameter containing the focal length and the magnification factor from the image size in mm to the imaging coordinates in pixels. Generally, the magnification factor can be known from the camera and the frame grabber manufacturer's specifications. The focal length can thus be calculated from k and the magnification factor. In fact, the points $[m'_{11}/m'_{31}, m'_{21}/m'_{31}]$ and $[m'_{12}/m'_{32}, m'_{22}/m'_{32}]$ are two vanishing points and the focal length can be deduced using geometry from the two orthogonal vanishing points in a single image [18].

(3) If both k_x and k_y are taken as k, then equation (15) can be rewritten as

$$(u_{hvi} - u_0)(u_{vvi} - u_0) + (v_{hvi} - v_0)(v_{vvi} - v_0) + k^2 = 0 \tag{20}$$

where $u_{hvi} = m'_{11i}/m'_{31i}$, $u_{vvi} = m'_{12i}/m'_{32i}$, $v_{hvi} = m'_{21i}/m'_{31i}$, $v_{vvi} = m'_{22i}/m'_{32i}$, are the image coordinates of the vanishing points.

Obviously, two groups of orthogonal parallels at one view can only provide one equation as given in (20). If there are three views or three groups of orthogonal parallels, we can deduce from (20)

$$\begin{cases}(u_{hv2}+u_{vv2}-u_{hv1}-u_{vv1})u_0+(v_{hv2}+v_{vv2}-v_{hv1}-v_{vv1})v_0 \\ =u_{hv2}u_{vv2}-u_{hv1}u_{vv1}+v_{hv2}v_{vv2}-v_{hv1}v_{vv1} \\ (u_{hv3}+u_{vv3}-u_{hv2}-u_{vv2})u_0+(v_{hv3}+v_{vv3}-v_{hv2}-v_{vv2})v_0 \\ =u_{hv3}u_{vv3}-u_{hv2}u_{vv2}+v_{hv3}v_{vv3}-v_{hv2}v_{vv2}\end{cases} \tag{21}$$

Then, u_0 and v_0 are computed from equation (21). k can be resolved from equation (19).

We can conclude that any three groups of orthogonal parallels with different poses are sufficient for the calibration of a camera with a three intrinsic parameter model. The method proposed by Bénallal *et al* [17] is a special case, which employs a cubic object to extract three orthogonal vanishing points and deduce the intrinsic parameters for a camera.

4 Pose detection

4.1 Camera pose determination

Parallels often exist in indoor environments, such as doors, windows and tables. It is natural to select parallel lines as reference. Consider the case of two groups of parallels, in which there are at least two lines in each group, and each line in a group is perpendicular to the lines in the other group. The world frame $\mathcal{W}$ is assigned to describe the poses of the lines. Its origin O_w can be selected at any place. Its X-axis X_w is set to be parallel to the lines in one group and its Y-axis Y_w parallel to the lines in the other group. The frame assignment is shown in Fig.2. The coordinates of point P_i are represented as $[x_{wi}, y_{wi}, z_{wi}]$ in the world frame. The four points P_1 to P_4 are arbitrarily selected on the two lines, i.e. two points for each line, for further pose computation in the camera frame.

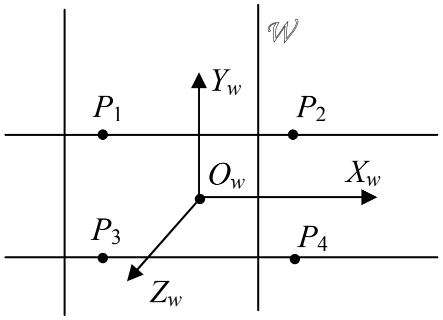


Fig. 2 The world frame of two groups of parallels

The origin of the camera frame is assigned at its optical centre. Z-axis is taken as the direction of its optical axis from the camera to the scene. X-axis is selected as the horizontal direction of its imaging plane from left to right. Two frames, $\mathbb{C}_1$ and $\mathbb{C}_2$, are given for cameras C_{a1} and C_{a2} in Fig.3, in which ${}^1T_2$ and ${}^1R_2$ are the translation and rotation from O_{c1} to O_{c2}. ${}^1H_2$, the transformation from $\mathbb{C}_1$ to $\mathbb{C}_2$, is expressed as (22) with homogeneous transformation.

$$ {}^1H_2 = \begin{bmatrix} {}^1R_2 & {}^1T_2 \\ 0 & 1 \end{bmatrix} \tag{22} $$

If ${}^1M_2$ is employed to denote the extrinsic parameters matrix from $\mathbb{C}_1$ to $\mathbb{C}_2$, then ${}^1H_2$ is the homogeneous form of matrix ${}^1M_2$.

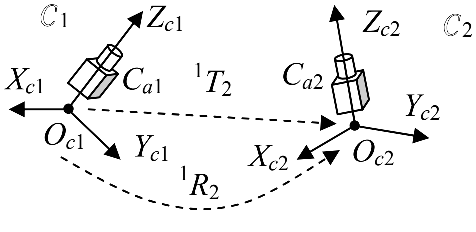


Fig.3 The camera frames assignment

Firstly, the pose of the parallels relative to the camera frame $\mathbb{C}_1$ is considered. According to the orthogonal constraints in rotation matrix, for a point P on line L_k, from equation (2), we have

$$\begin{cases} {}^1o_{wx}x_{c1k} + {}^1o_{wy}y_{c1k} + {}^1o_{wz}z_{c1k} = y_{wk} + {}^1o_{wx}\,{}^1p_{wx} + {}^1o_{wy}\,{}^1p_{wy} + {}^1o_{wz}\,{}^1p_{wz} \\ {}^1a_{wx}x_{c1k} + {}^1a_{wy}y_{c1k} + {}^1a_{wz}z_{c1k} = z_{wk} + {}^1a_{wx}\,{}^1p_{wx} + {}^1a_{wy}\,{}^1p_{wy} + {}^1a_{wz}\,{}^1p_{wz} \end{cases} \tag{23}$$

where y_{wk} is the coordinates of line L_k at the Y-axis, z_{wk} is that at the Z-axis in the frame $\mathbb{W}$. The subscript k represents a point on line L_k, subscript $c1$ represents the data in $\mathbb{C}_1$, subscript w means the frame $\mathbb{W}$, the combination of superscript 1 and subscript w indicates the data of extrinsic parameters from $\mathbb{C}_1$ to $\mathbb{W}$.

Let

$$\begin{cases} {}^1A_k = y_{wk} + {}^1o_{wx}\,{}^1p_{wx} + {}^1o_{wy}\,{}^1p_{wy} + {}^1o_{wz}\,{}^1p_{wz} \\ {}^1B_k = z_{wk} + {}^1a_{wx}\,{}^1p_{wx} + {}^1a_{wy}\,{}^1p_{wy} + {}^1a_{wz}\,{}^1p_{wz} \end{cases} \tag{24}$$

For any points in the two groups of lines, $z_{c1k} \neq 0$ are satisfied. Therefore, equation (23) can be rewritten as

$$\frac{{}^1o_{wx}x'_{1k} + {}^1o_{wy}y'_{1k} + {}^1o_{wz}}{{}^1a_{wx}x'_{1k} + {}^1a_{wy}y'_{1k} + {}^1a_{wz}} = {}^1C_k \tag{25}$$

where ${}^1C_k = {}^1A_k / {}^1B_k$, $x'_{1k} = x_{c1k} / z_{c1k}$ and $y'_{1k} = y_{c1k} / z_{c1k}$.

All points on the line L_k have the same coordinate $y_w = y_{wk}$ at the Y-axis, and $z_w = z_{wk}$ at the Z-axis. The extrinsic parameters for any point on the line are also kept the same. Therefore, it is easy to conclude from equation (24) that ${}^1A_k$, ${}^1B_k$ and ${}^1C_k$ are constants for the line L_k. For two arbitrary points on the line L_k, such as points P_i and P_j, equation (26) is obtained via applying them to equation (25). Its simplified form is equation (27), which results from its simplification using the orthogonal constraints of rotation matrix in ${}^1M_w$.

$$\frac{{}^1o_{wx}x'_{1ki}+{}^1o_{wy}y'_{1ki}+{}^1o_{wz}}{{}^1a_{wx}x'_{1ki}+{}^1a_{wy}y'_{1ki}+{}^1a_{wz}}=\frac{{}^1o_{wx}x'_{1kj}+{}^1o_{wy}y'_{1kj}+{}^1o_{wz}}{{}^1a_{wx}x'_{1kj}+{}^1a_{wy}y'_{1kj}+{}^1a_{wz}} \tag{26}$$

$${}^1n_{wx}(y'_{1ki}-y'_{1kj})+{}^1n_{wy}(x'_{1kj}-x'_{1ki})+{}^1n_{wz}(x'_{1ki}y'_{1kj}-x'_{1kj}y'_{1ki})=0 \tag{27}$$

Any two points on a line that is parallel to the X-axis in the frame $\mathcal{W}$ satisfy equation (27). Therefore, k equations can be obtained for one camera from k parallel lines. According to the main direction of the lines in image space, one parameter in ${}^1\vec{n}_w$ can be removed from equation (27) with the reduction of a fraction. As a general case, assume ${}^1n_{wx}\neq 0$. We have

$$\begin{cases} MN=L \\ M=\begin{bmatrix} x'_{11j}-x'_{11i} & x'_{11i}y'_{11j}-x'_{11j}y'_{11i} \\ \vdots & \vdots \\ x'_{1kj}-x'_{1ki} & x'_{1ki}y'_{1kj}-x'_{1kj}y'_{1ki} \end{bmatrix}, N=\begin{bmatrix} {}^1n'_{wy} \\ {}^1n'_{wz} \end{bmatrix}, L=\begin{bmatrix} y'_{11j}-y'_{11i} \\ \vdots \\ y'_{1kj}-y'_{1ki} \end{bmatrix} \end{cases} \tag{28}$$

where ${}^1n'_{wy}={}^1n_{wy}/{}^1n_{wx}$ and ${}^1n'_{wz}={}^1n_{wz}/{}^1n_{wx}$.

Notice that x'_{1hi}, y'_{1hi}, x'_{1hj} and y'_{1hj} (h=1,…,k) can be obtained from equation (1) according to the imaging coordinates [u, v]. Therefore, N can be resolved with Least Square Method.

$$N=(M^TM)^{-1}L \tag{29}$$

Then, ${}^1\vec{n}_w$ can be calculated with the constraint $\left\|{}^1\vec{n}_w\right\|=1$.

The vector ${}^1\vec{o}_w$ can be obtained with a similar procedure. With the orthogonal constraint, ${}^1\vec{a}_w$ is obtained as ${}^1\vec{a}_w={}^1\vec{n}_w\times{}^1\vec{o}_w$. The pose of the frame $\mathcal{W}$ relative to the frame $\mathcal{C}_1$ is represented as

${}^{1}R_{w}=\left[{}^{1}\vec{n}_{w}\quad {}^{1}\vec{o}_{w}\quad {}^{1}\vec{a}_{w}\right]$. ${}^{2}R_{w}$, i.e. the pose of the frame $\mathcal{W}$ relative to $\mathbb{C}_2$, can also be deduced with a similar method. The relative pose from $\mathbb{C}_1$ to $\mathbb{C}_2$ is calculated with rotation transformation, i.e. ${}^{1}R_{2}={}^{1}R_{w}({}^{2}R_{w})^{-1}$.

4.2 Pose detection for an object

4.2.1 Pose detection for a line

Any two points can be selected as features from a line L_k in the images of the two cameras. Here the feature matching of the points for the two cameras is not necessary. As shown in Fig.4, P_{11} and P_{12} are two feature points selected from a line for camera C_{a1}; P_{21} and P_{22} are for camera C_{a2}. The vector formed by P_{11} and P_{12} is denoted as ${}^{1}\vec{L}_{p12}$ and ${}^{2}\vec{L}_{p12}$ in $\mathbb{C}_1$ and $\mathbb{C}_2$ separately. The position vector of a point P_i is represented as $\vec{P}_{c1i}$ and $\vec{P}_{c2i}$ in $\mathbb{C}_1$ and $\mathbb{C}_2$. ${}^{1}\vec{P}_{c2i}$ is the representation of $\vec{P}_{c2i}$ in $\mathbb{C}_1$. $\vec{P}_{c1i}$ and $\vec{P}_{c2i}$ can be calculated from equation (1) according to the imaging coordinates of P_i in $\mathbb{C}_1$ and $\mathbb{C}_2$, as given in (30).

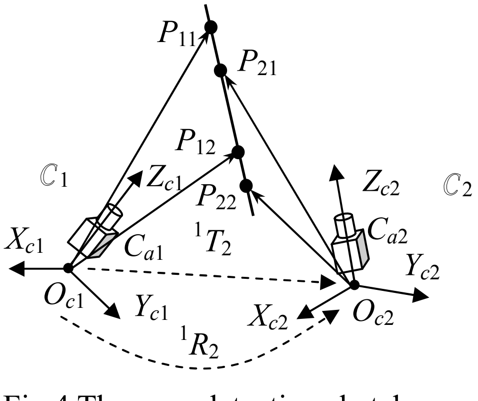


Fig.4 The pose detection sketch map

$$\begin{cases} \vec{P}_{c1i} = \left[x_{c1i}/z_{c1i} \quad y_{c1i}/z_{c1i} \quad 1\right]^T = M_{in1}^{-1}\left[u_{1i} \quad v_{1i} \quad 1\right]^T \\ \vec{P}_{c2i} = \left[x_{c2i}/z_{c2i} \quad y_{c2i}/z_{c2i} \quad 1\right]^T = M_{in2}^{-1}\left[u_{2i} \quad v_{2i} \quad 1\right]^T \end{cases} \tag{30}$$

where M_{in1} is the intrinsic parameter matrix of camera C_{a1} and M_{in2} is that of camera C_{a2}, $[u_{1i}, v_{1i}]$ and $[u_{2i}, v_{2i}]$ are the imaging coordinates of P_i in $\mathbb{C}_1$ and $\mathbb{C}_2$.

The three vectors, ${}^1\vec{L}_{p12}$, ${}^1\vec{P}_{c11}$ and ${}^1\vec{P}_{c12}$, are linearly correlated as they form a vector triangle in a plane. Therefore

$$\begin{vmatrix} P_{11x} & P_{12x} & {}^1L_{12x} \\ P_{11y} & P_{12y} & {}^1L_{12y} \\ P_{11z} & P_{12z} & {}^1L_{12z} \end{vmatrix} = 0 \tag{31}$$

where ${}^1\vec{P}_{c11} = \vec{P}_{c11} = \left[P_{11x} \quad P_{11y} \quad P_{11z}\right]^T$, ${}^1\vec{P}_{c12} = \vec{P}_{c12} = \left[P_{12x} \quad P_{12y} \quad P_{12z}\right]^T$, ${}^1\vec{L}_{p12} = \left[{}^1L_{12x} \quad {}^1L_{12y} \quad {}^1L_{12z}\right]^T$. The evaluations of ${}^1\vec{P}_{c11}$ and ${}^1\vec{P}_{c12}$ are as shown in expression (30).

Equation (31) can be rewritten as

$$\begin{aligned} &(P_{11y}P_{12z} - P_{12y}P_{11z})^1L_{12x} + (P_{12x}P_{11z} - P_{11x}P_{12z})^1L_{12y} \\ &+ (P_{11x}P_{12y} - P_{12x}P_{11y})^1L_{12z} = 0 \end{aligned} \tag{32}$$

On the other hand, ${}^1\vec{P}_{c2i}$ is transformed from $\vec{P}_{c2i}$ with rotation transformation ${}^1R_2$,

$${}^1\vec{P}_{c2i} = {}^1R_2\vec{P}_{c2i} = \begin{bmatrix} {}^1n_{2x}x_{c2i}/z_{c2i} + {}^1o_{2x}y_{c2i}/z_{c2i} + {}^1a_{2x} \\ {}^1n_{2y}x_{c2i}/z_{c2i} + {}^1o_{2y}y_{c2i}/z_{c2i} + {}^1a_{2y} \\ {}^1n_{2z}x_{c2i}/z_{c2i} + {}^1o_{2z}y_{c2i}/z_{c2i} + {}^1a_{2z} \end{bmatrix} = \begin{bmatrix} {}^1P_{2ix} \\ {}^1P_{2iy} \\ {}^1P_{2iz} \end{bmatrix} \tag{33}$$

In the same way, the following can be deduced with three linear correlation vectors, ${}^1\vec{L}_{p12}$, ${}^1\vec{P}_{c21}$ and ${}^1\vec{P}_{c22}$

$$\begin{aligned} &({}^1P_{21y}{}^1P_{22z} - {}^1P_{22y}{}^1P_{21z})^1L_{12x} + ({}^1P_{22x}{}^1P_{21z} - {}^1P_{21x}{}^1P_{22z})^1L_{12y} \\ &+ ({}^1P_{21x}{}^1P_{22y} - {}^1P_{22x}{}^1P_{21y})^1L_{12z} = 0 \end{aligned} \tag{34}$$

From equations (32) and (34), ${}^{1}\vec{L}_{p12}$ can be determined through the constraint $\left\| {}^{1}\vec{L}_{p12} \right\| = 1$. Then ${}^{2}\vec{L}_{p12}$ becomes

$$ {}^{2}\vec{L}_{p12} = ({}^{1}R_{2})^{-1}\ {}^{1}\vec{L}_{p12} \tag{35} $$

4.2.2 Pose detections for a plane and rigid object

For a plane, at least three points are selected for its norm detection. Any two points among them form a vector, whose pose can be determined as described above. There are only two independent vectors among the three vectors formed with the three points. With the poses of the independent vectors, the pose of the plane norm can be obtained with vector product.

Furthermore, the pose detection based on vector measurement can be extended to pose determination for a rigid object. For example, the pose of a cuboid can be detected from three vectors of the edges with the above method. The details are omitted here.

5 A new visual control scheme based on pose estimation

5.1 The system configuration

A task for an industrial robot to approach an object with a desired pose for its end-effector is considered here. The system configuration for the task is sketched in Fig.5. Suppose that the object is rigid with parallel line edges. No further information about it is known. The two cameras, with frames $\mathbb{C}_1$ and $\mathbb{C}_2$, are located where the object and the end-effector are all in their field of view. The axes of frame $\mathbb{C}_1$ are parallel to the axes of frame $\mathcal{W}_r$, the world frame of the robot. That is, when the end-efector is moved in the direction of X-axis of the frame $\mathcal{W}_r$, the image coordinates of the end-effector vary only in the horizontal direction. When the end-efector is moved in the direction of Z-axis of the frame $\mathcal{W}_r$, its image coordinates vary only in the vertical direction. In addition, $\mathbb{E}$ and $\mathcal{W}_o$ represent the end-effector frame and the object frame.

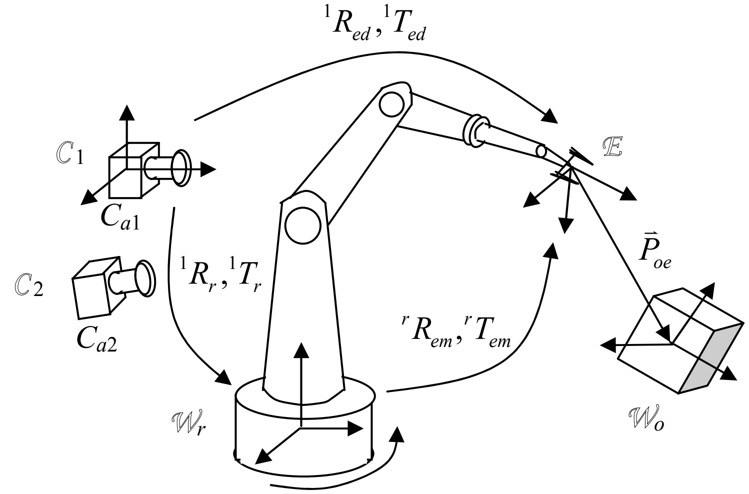


Fig.5 The configuration of visual control for a manipulator to approach an object based on pose detection

5.2 The control strategy

In frame $\mathbb{C}_1$, the poses of the sidelines of the rectangle surface of the object are detected with the method in Section 4.2. Then the norm of the surface is calculated using vector product of the two neighbor sidelines. Now, ${}^1R_o$, the pose of the frame $\mathcal{W}_o$ expressed in $\mathbb{C}_1$, is obtained. ${}^1R_{ed}$, the desired pose of the end-effector in $\mathbb{C}_1$, is determined from ${}^1R_o$. ${}^rR_{em}$, the currently measured pose of the end-effector in $\mathcal{W}_r$, can be given from the robot's controller.

Assume that the pose of frame $\mathcal{W}_r$ is ${}^1R_r$ in the frame $\mathcal{C}_1$. Frame $\mathcal{W}_r$ is selected as the reference. Assume that the pose adjustment is ${}^rR_{ea}$ in the frame $\mathcal{W}_r$. Based on frame $\mathcal{W}_r$, equation (36) is deduced from frame transforms. Then ${}^rR_{ea}$ is obtained as given in (37).

$$({}^1R_r)^{-1}\ {}^1R_{ed} = {}^rR_{ea}\ {}^rR_{em} \tag{36}$$

$${}^rR_{ea} = ({}^1R_r)^{-1}\ {}^1R_{ed}\,({}^rR_{em})^{-1} \tag{37}$$

In fact, ${}^rR_{ea}$ is the pose error between the desired and measured poses. It can be represented as a rotation around an axis ${}^r\vec{f}$ with an angle ${}^r\theta$, as expressed in equation (38). $k_\theta\,{}^r\theta$ is employed as the actual angle adjustment value in one control step for pose control. The actual pose adjustment control value in one control step is computed in equation (39).

$${}^rR_{ea} = Rot({}^r\vec{f}, {}^r\theta) \tag{38}$$

$${}^rR_{ec} = Rot({}^r\vec{f}, k_\theta\,{}^r\theta), \quad 0 < k_\theta < 1 \tag{39}$$

where ${}^rR_{ec}$ is the actual pose adjustment control value in one control step, k_θ is the factor of pose adjustment.

To decouple the pose and position adjustments, the position of the end-effector is assumed to be fixed during the pose adjustment. The position and pose of the end-effector in frame $\mathcal{W}_r$ can be computed via the kinematics combined with the measured values of the joint angles of the industrial robot. Suppose that the measured position and pose of the end-effector is

$$
{}^{r}H_{em} = \begin{bmatrix} {}^{r}R_{em} & {}^{r}T_{em} \\ 0 & 1 \end{bmatrix} \tag{40}
$$

The translation to compensate for the offset caused by pose adjustment is

$$
{}^{r}T_{e1} =-{}^{1}R_{ec}\,{}^{r}T_{em} \tag{41}
$$

At the beginning of the approaching task, the distance between the end-effector and the object is unknown. Therefore, the translation to approach the object can be evaluated with the constant step length along the direction from the end-effector to the object.

$$
{}^{r}T_{e2} = k_s({}^{1}R_r)^{-1}\ {}^{1}\vec{P}'_{oe}, \quad 0 < k_s < 1 \tag{42}
$$

where ${}^{1}\vec{P}'_{oe}$ is the unit vector of $\vec{P}_{oe}$ in frame $\mathcal{C}_1$. $\vec{P}_{oe}$ is a vector from the origin of the frame $\mathcal{E}$. to the origin of the frame $\mathcal{W}_o$, which can be measured with the method in Section 4.2. k_s is the position step factor at the beginning of the task.

The distance moved by the end-effector can be computed with the position of the end-effector, which is read from the position controller of the robot. Assume that the end-effector is moved in the direction of vector $\vec{P}_{oe}$. After the end-effector is moved at least two steps, the distance from the end-effector to the object can be estimated with the cross ratio invariance. Then

$$
\begin{cases}
\dfrac{(d_{em}^{i-1}+d_{em}^{i})/d_{em}^{i}}{(d_{em}^{i-1}+d_{em}^{i}+d_{ed1}^{i})/(d_{em}^{i}+d_{ed1}^{i})} = \dfrac{(r_{e1}^{i-1}+r_{e1}^{i})/r_{e1}^{i}}{(r_{e1}^{i-1}+r_{e1}^{i}+r_{de1}^{i})/(r_{e1}^{i}+r_{de1}^{i})} \\
\dfrac{(d_{em}^{i-1}+d_{em}^{i})/d_{em}^{i}}{(d_{em}^{i-1}+d_{em}^{i}+d_{ed2}^{i})/(d_{em}^{i}+d_{ed2}^{i})} = \dfrac{(r_{e2}^{i-1}+r_{e2}^{i})/r_{e2}^{i}}{(r_{e2}^{i-1}+r_{e2}^{i}+r_{de2}^{i})/(r_{e2}^{i}+r_{de2}^{i})}
\end{cases} \tag{43}
$$

where d^{i}_{ed1} is the estimated distance between the target and tool points using the camera C_{a1} at the i-th sampling, d^i_{ed2} is the estimated distance via camera C_{a2}, r^i_{de1} is the image distance between the target and tool points in camera C_{a1} at the i-th sampling step, r^i_{de2} is the image distance in camera C_{a2}, d^i_{em} is the distance of the end-effector moved between i-1-th and i-th sampling, r^i_{e1} and r^i_{e2} are the image distances of the end-effector moved in the image spaces of cameras C_{a1} and C_{a2} between i-1-th and i-th sampling, r^i_{e1} and r^i_{e2} are corresponding to d^i_{em} (i>2).

$$\begin{cases} r^i_{e1} = \sqrt{(u^i_{e1} - u^{i-1}_{e1})^2 + (v^i_{e1} - v^{i-1}_{e1})^2} \\ r^i_{de1} = \sqrt{(u_{d1} - u^i_{e1})^2 + (v_{d1} - v^i_{e1})^2} \\ r^i_{e2} = \sqrt{(u^i_{e2} - u^{i-1}_{e2})^2 + (v^i_{e2} - v^{i-1}_{e2})^2} \\ r^i_{de2} = \sqrt{(u_{d2} - u^i_{e2})^2 + (v_{d2} - v^i_{e2})^2} \end{cases} \tag{44}$$

where $[u_{d1}, v_{d1}]$ and $[u_{d2}, v_{d2}]$ are the image coordinates of the object in cameras C_{a1} and C_{a2}, $[u^i_{e1}, v^i_{e1}]$ and $[u^i_{e2}, v^i_{e2}]$ are the image coordinates of the end-effector in cameras C_{a1} and C_{a2} at i-th sampling.

If r^i_{e1} is zero, d^i_{ed1} cannot be estimated from equation (43). It is the same for d^i_{ed2} if r^i_{e2} is zero. If only one image distance is nonzero, r^i_{e1} or r^i_{e2}, then the only distance d^i_{ed1} or d^i_{ed2} estimated from equation (43) is taken as the distance d^i_{ed}. If both image distances are nonzero, then the average of d^i_{ed1} and d^i_{ed2} is taken as the distance d^i_{ed}. If both image distances are zero, then the distance d^i_{ed} is zero. This means that the end-effector has reached the target. d^i_{ed} is the estimated distance between the target and tool points at the i-th sampling.

With the errors d^i_{ed} and the vector ${}^1\vec{P}_{oe}$, the errors e^i_{ed} between the target and tool points are expressed as (45), and the approaching control law with PID algorithm is given in (46).

$$\begin{bmatrix} e^i_{xed} & e^i_{yed} & e^i_{zed} \end{bmatrix}^T = d^i_{ed}\,({}^1R_r)^{-1}\;{}^1\vec{P}'_{eo} \tag{45}$$

$$ {}^rT_{e2} = \begin{bmatrix} \Delta x^i_{ce} \\ \Delta y^i_{ce} \\ \Delta z^i_{ce} \end{bmatrix} = K_p \begin{bmatrix} e^i_{xed} \\ e^i_{yed} \\ e^i_{zed} \end{bmatrix} + K_i \begin{bmatrix} \sum\limits_{j=1}^{i} e^i_{xed} \\ \sum\limits_{j=1}^{i} e^i_{yed} \\ \sum\limits_{j=1}^{i} e^i_{zed} \end{bmatrix} + K_d \begin{bmatrix} e^i_{xed} - e^{i-1}_{xed} \\ e^i_{yed} - e^{i-1}_{yed} \\ e^i_{zed} - e^{i-1}_{zed} \end{bmatrix} \tag{46}$$

where Δx^i_{ce} is the adjustment value of the end-effector in the direction of X-axis, Δy^i_{ce} is that in the direction of Y-axis, Δz^i_{ce} is that in the direction of Z-axis in frame $\mathcal{W}_r$. K_p, K_i, K_d are PID gain matrixes.

The adjustments to control the end-effector to approach the object are formed as equation (47) in frame $\mathcal{W}_r$, the world frame of the robot.

$${}^rH_{ec} = \begin{bmatrix} {}^rR_{ec} & {}^rT_{e1} + {}^rT_{e2} \\ 0 & 1 \end{bmatrix} \tag{47}$$

where ${}^rH_{ec}$ is the actual adjustment control value in one control step.

6 Sensitivity analysis

6.1 Errors in line pose detection

In the parameters of a line, there exists at least one parameter of non-zero value. Without losing generality, suppose ${}^1L_{12z}\neq 0$. Considering $P_{11z}=P_{12z}=1$, we can deduce from equation (32) and (34)

$$\begin{cases} (P_{11y} - P_{12y})L_x + (P_{12x} - P_{11x})L_y = P_{12x}P_{11y} - P_{11x}P_{12y} \\ ({}^1P_{21y}\,{}^1P_{22z} - {}^1P_{22y}\,{}^1P_{21z})L_x + ({}^1P_{22x}\,{}^1P_{21z} - {}^1P_{21x}\,{}^1P_{22z})L_y = {}^1P_{22x}\,{}^1P_{21y} - {}^1P_{21x}\,{}^1P_{22y} \end{cases} \tag{48}$$

where $L_x = {}^1L_{12x}/{}^1L_{12z}$, $L_y = {}^1L_{12y}/{}^1L_{12z}$.

The solution of L_x and L_y can be expressed as

$$\begin{cases} L_x = (a_3 b_2 - a_2 b_3)/(a_1 b_2 - a_2 b_1) \\ L_y = (a_1 b_3 - a_3 b_1)/(a_1 b_2 - a_2 b_1) \end{cases} \tag{49}$$

where

$$\begin{cases} a_1 = P_{11y} - P_{12y}, a_2 = P_{12x} - P_{11x}, a_3 = P_{12x} P_{11y} - P_{11x} P_{12y} \\ b_1 = {}^1P_{21y}\, {}^1P_{22z} - {}^1P_{22y}\, {}^1P_{21z}, b_2 = {}^1P_{22x}\, {}^1P_{21z} - {}^1P_{21x}\, {}^1P_{22z}, b_3 = {}^1P_{22x}\, {}^1P_{21y} - {}^1P_{21x}\, {}^1P_{22y} \end{cases} \tag{50}$$

From equation (49), the differences ΔL_x and ΔL_y can be derived as in (51). The derivatives $\partial L_x / \partial a_k$, $\partial L_x / \partial b_k$, $\partial L_y / \partial a_k$ and $\partial L_y / \partial b_k$ are given in (52). For $\partial a_k / \partial u_{ij}$ and $\partial b_k / \partial u_{ij}$, most terms are zero. To simplify the discussion, a special case is considered that the relative poses between two cameras are unit matrices. In such a case, ${}^1P_{11z}={}^1P_{12z}=1$. The non-zero terms for $\partial a_k / \partial u_{ij}$ and $\partial b_k / \partial u_{ij}$ are given in (53), while the other terms are zero.

$$\begin{cases} \Delta L_x = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{3}(\frac{\partial L_x}{\partial a_k}\frac{\partial a_k}{\partial u_{ij}} + \frac{\partial L_x}{\partial b_k}\frac{\partial b_k}{\partial u_{ij}})\Delta u_{ij} + \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{3}(\frac{\partial L_x}{\partial a_k}\frac{\partial a_k}{\partial v_{ij}} + \frac{\partial L_x}{\partial b_k}\frac{\partial b_k}{\partial v_{ij}})\Delta v_{ij} \\ \Delta L_y = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{3}(\frac{\partial L_y}{\partial a_k}\frac{\partial a_k}{\partial u_{ij}} + \frac{\partial L_y}{\partial b_k}\frac{\partial b_k}{\partial u_{ij}})\Delta u_{ij} + \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{3}(\frac{\partial L_y}{\partial a_k}\frac{\partial a_k}{\partial v_{ij}} + \frac{\partial L_y}{\partial b_k}\frac{\partial b_k}{\partial v_{ij}})\Delta v_{ij} \end{cases} \tag{51}$$

$$\begin{cases} \frac{\partial L_x}{\partial a_1} = \frac{-L_x b_2}{a_1 b_2 - a_2 b_1}, \frac{\partial L_x}{\partial a_2} = \frac{-L_y b_2}{a_1 b_2 - a_2 b_1}, \frac{\partial L_x}{\partial a_3} = \frac{b_2}{a_1 b_2 - a_2 b_1}, \frac{\partial L_x}{\partial b_1} = \frac{L_x a_2}{a_1 b_2 - a_2 b_1} \\ \frac{\partial L_x}{\partial b_2} = \frac{L_y a_2}{a_1 b_2 - a_2 b_1}, \frac{\partial L_x}{\partial b_3} = \frac{-a_2}{a_1 b_2 - a_2 b_1}, \frac{\partial L_y}{\partial a_1} = \frac{L_x b_1}{a_1 b_2 - a_2 b_1}, \frac{\partial L_y}{\partial a_2} = \frac{L_y b_1}{a_1 b_2 - a_2 b_1} \\ \frac{\partial L_y}{\partial a_3} = \frac{-b_1}{a_1 b_2 - a_2 b_1}, \frac{\partial L_y}{\partial b_1} = \frac{-L_x a_1}{a_1 b_2 - a_2 b_1}, \frac{\partial L_y}{\partial b_2} = \frac{-L_y a_1}{a_1 b_2 - a_2 b_1}, \frac{\partial L_y}{\partial b_3} = \frac{a_1}{a_1 b_2 - a_2 b_1} \end{cases} \tag{52}$$

$$\begin{cases} \frac{\partial a_1}{\partial v_{11}} = \frac{1}{k_{y1}}, \frac{\partial a_1}{\partial v_{12}} = -\frac{1}{k_{y1}}, \frac{\partial a_2}{\partial u_{11}} = -\frac{1}{k_{x1}}, \frac{\partial a_2}{\partial u_{12}} = \frac{1}{k_{x1}}, \frac{\partial a_3}{\partial u_{11}} = -\frac{P_{12y}}{k_{x1}}, \frac{\partial a_3}{\partial u_{12}} = \frac{P_{11y}}{k_{x1}}, \frac{\partial a_3}{\partial v_{11}} = \frac{P_{12x}}{k_{y1}}, \frac{\partial a_3}{\partial v_{12}} = -\frac{P_{11x}}{k_{y1}} \\ \frac{\partial b_1}{\partial v_{21}} = \frac{1}{k_{y2}}, \frac{\partial b_1}{\partial v_{22}} = -\frac{1}{k_{y2}}, \frac{\partial b_2}{\partial u_{21}} = -\frac{1}{k_{x2}}, \frac{\partial b_2}{\partial u_{22}} = \frac{1}{k_{x2}}, \frac{\partial b_3}{\partial u_{21}} = -\frac{{}^1P_{22y}}{k_{x2}}, \frac{\partial b_3}{\partial u_{22}} = \frac{{}^1P_{21y}}{k_{x2}}, \frac{\partial b_3}{\partial v_{21}} = \frac{{}^1P_{22x}}{k_{y2}}, \frac{\partial b_3}{\partial v_{22}} = -\frac{{}^1P_{21x}}{k_{y2}} \end{cases} \tag{53}$$

Applying formula (52) and (53) to (51), then ΔL_x and ΔL_y are obtained as

$$\begin{cases}\Delta L_x = \dfrac{1}{a_1 b_2 - a_2 b_1}\Bigg[-L_x b_2 \left(\dfrac{\Delta v_{11} - \Delta v_{12}}{k_{y1}}\right) + L_x a_2 \left(\dfrac{\Delta v_{21} - \Delta v_{22}}{k_{y2}}\right) - L_y b_2 \left(\dfrac{\Delta u_{12} - \Delta u_{11}}{k_{x1}}\right) + L_y a_2 \left(\dfrac{\Delta u_{22} - \Delta u_{21}}{k_{x2}}\right) \\ + b_2 \left(\dfrac{P_{11y}\Delta u_{12} - P_{12y}\Delta u_{11}}{k_{x1}}\right) - a_2 \left(\dfrac{{}^1P_{21y}\Delta u_{22} - {}^1P_{22y}\Delta u_{21}}{k_{x2}}\right) + b_2 \left(\dfrac{P_{12x}\Delta v_{11} - P_{11x}\Delta v_{12}}{k_{y1}}\right) - a_2 \left(\dfrac{{}^1P_{22x}\Delta v_{21} - {}^1P_{21x}\Delta v_{22}}{k_{y2}}\right)\Bigg] \\ \Delta L_y = \dfrac{1}{a_1 b_2 - a_2 b_1}\Bigg[L_x b_1 \left(\dfrac{\Delta v_{11} - \Delta v_{12}}{k_{y1}}\right) - L_x a_1 \left(\dfrac{\Delta v_{21} - \Delta v_{22}}{k_{y2}}\right) + L_y b_1 \left(\dfrac{\Delta u_{12} - \Delta u_{11}}{k_{x1}}\right) - L_y a_1 \left(\dfrac{\Delta u_{22} - \Delta u_{21}}{k_{x2}}\right) \\ - b_1 \left(\dfrac{P_{11y}\Delta u_{12} - P_{12y}\Delta u_{11}}{k_{x1}}\right) + a_1 \left(\dfrac{{}^1P_{21y}\Delta u_{22} - {}^1P_{22y}\Delta u_{21}}{k_{x2}}\right) - b_1 \left(\dfrac{P_{12x}\Delta v_{11} - P_{11x}\Delta v_{12}}{k_{y1}}\right) + a_1 \left(\dfrac{{}^1P_{22x}\Delta v_{21} - {}^1P_{21x}\Delta v_{22}}{k_{y2}}\right)\Bigg] \end{cases} \quad (54)$$

Generally, the parameters k_x and k_y are very close to each other. For the two cameras, k_{x1}, k_{y1}, k_{x2} and k_{y2} are taken as k to obtain approximate values of ΔL_x and ΔL_y.

$$\begin{cases}\Delta L_x \approx \dfrac{1}{(a_1 b_2 - a_2 b_1)k}\big[(L_y - P_{12y})b_2\Delta u_{11} + (P_{11y} - L_y)b_2\Delta u_{12} + ({}^1P_{22y} - L_y)a_2\Delta u_{21} + (L_y - {}^1P_{21y})a_2\Delta u_{22} \\ + (P_{12x} - L_x)b_2\Delta v_{11} + (L_x - P_{11x})b_2\Delta v_{12} + (L_x - {}^1P_{22x})a_2\Delta v_{21} + ({}^1P_{21x} - L_x)a_2\Delta v_{22}\big] \\ \Delta L_y \approx \dfrac{1}{(a_1 b_2 - a_2 b_1)k}\big[(P_{12y} - L_y)b_1\Delta u_{11} + (L_y - P_{11y})b_1\Delta u_{12} + (L_y - {}^1P_{22y})a_1\Delta u_{21} + ({}^1P_{21y} - L_y)a_1\Delta u_{22} \\ + (L_x - P_{12x})b_1\Delta v_{11} + (P_{11x} - L_x)b_1\Delta v_{12} + ({}^1P_{22x} - L_x)a_1\Delta v_{21} + (L_x - {}^1P_{21x})a_1\Delta v_{22}\big] \end{cases} \quad (55)$$

The line pose ${}^1\vec{L}_{p12}$ can be determined with L_x and L_y as

$$\begin{cases} {}^1L_{12x} = L_x / \sqrt{L_x^2 + L_y^2 + 1} \\ {}^1L_{12y} = L_y / \sqrt{L_x^2 + L_y^2 + 1} \\ {}^1L_{12z} = 1 / \sqrt{L_x^2 + L_y^2 + 1} \end{cases} \quad (56)$$

Therefore, the errors in line pose caused by ΔL_x and ΔL_y are given as

$$\begin{cases} dL_{12x} = [(1 + L_y^2)\Delta L_x - L_x L_y \Delta L_y]/(L_x^2 + L_y^2 + 1)^{3/2} \\ dL_{12y} = [(1 + L_x^2)\Delta L_y - L_x L_y \Delta L_x]/(L_x^2 + L_y^2 + 1)^{3/2} \\ dL_{12z} = -(L_x \Delta L_x + L_y \Delta L_y)/(L_x^2 + L_y^2 + 1)^{3/2} \end{cases} \quad (57)$$

where dL_{12x}, dL_{12y} and dL_{12z} are the errors in the line pose.

Discussion:

Case 1: Δu_{11}=Δu_{12}=Δu_{21}=Δu_{22}=Δv_{11}=Δv_{12}=Δv_{21}=Δv_{21}=e_{uv}. From formula (55), the errors ΔL_x and ΔL_y are computed as given in (58). When e_{uv}=1 and k=1000, the errors are about 0.001.

$$\begin{cases} \Delta L_x \approx e_{uv} / k \\ \Delta L_y \approx e_{uv} / k \end{cases} \tag{58}$$

Case 2: Δu_{11}=-Δu_{12}=e_u, Δu_{21}=-Δu_{22}=-e_u, Δv_{11}=-Δv_{12}=e_v, Δv_{21}=-Δv_{22}=-e_v. This is a extreme case that the errors in the feature points in image space are opposite in directions. In this case, the errors ΔL_x and ΔL_y will reach the maximum values

$$\begin{cases} \Delta L_x \approx \dfrac{1}{(a_1 b_2 - a_2 b_1)k} \Big[2L_y (a_2 + b_2) e_u - (P_{12y} + P_{11y}) b_2 e_u - ({}^1P_{22y} + {}^1P_{21y}) a_2 e_u \\ \qquad + (P_{12x} + P_{11x}) b_2 e_v - 2L_x (a_2 + b_2) e_v + ({}^1P_{21x} + {}^1P_{22x}) a_2 e_v \Big] \\ \Delta L_y \approx \dfrac{1}{(a_1 b_2 - a_2 b_1)k} \Big[-2L_y (a_1 + b_1) e_u + (P_{12y} + P_{11y}) b_1 e_u + ({}^1P_{22y} + {}^1P_{21y}) a_1 e_u \\ \qquad + -(P_{12x} + P_{11x}) b_1 e_v + 2L_x (a_1 + b_1) e_v - ({}^1P_{21x} + {}^1P_{22x}) a_1 e_v \Big] \end{cases} \tag{59}$$

The precision in calibrating the intrinsic parameters of the two cameras is quite satisfactory. For example, for camera C_{a1}, we have k_{x1}=2499.9, k_{y1}=2364.1, u_{01}=367.4, v_{01}=285.2 and for Camera C_{a2}, k_{x2}=2478.1, k_{y2}=2352.6, u_{02}=374.1, v_{02}=261.5. The feature points extracted from the line are [52, 247] and [221, 194] in camera C_{a1}, and [400, 242] and [566, 186] in camera C_{a2}. According to formula (50), the temporary variables are computed as a_1=0.0224, a_2=0.0676, a_3=-0.0039, b_1=0.0238, b_2=0.0670, b_3=-0.0003. The solution of L_x and L_y are computed according to (49), the results are L_x=2.2518 and L_y=-0.8048. The obtained line pose is ${}^1L_{12x}$=0.8688, ${}^1L_{12y}$=-0.3105, ${}^1L_{12z}$=0.3858. If the errors in feature points are limited to 0.1 pixel, the errors calculated from (59) are ΔL_x =0.3064 and ΔL_y=-0.1053. The maximum errors in line pose will be dL_{12x}=0.0180, dL_{12x}=-0.0048, dL_{12x}=-0.0445. With Hough transform and Least Square Method, the accuracy of feature points in line can be improved further. Then the errors in the line pose estimation will be reduced proportionally.

6.2 The errors in estimated distance

Equation (43) on cross ratio invariance can be rewritten as

$$(d_{em}^{i-1}+d_{em}^{i})(d_{em}^{i}+d_{ed1}^{i})(r_{e1}^{i-1}+r_{e1}^{i}+r_{de1}^{i})r_{e1}^{i}-(d_{em}^{i-1}+d_{em}^{i}+d_{ed1}^{i})d_{em}^{i}(r_{e1}^{i-1}+r_{e1}^{i})(r_{e1}^{i}+r_{de1}^{i})=0 \tag{60}$$

Let F denote the left side of equation (60). The derivatives of F with respect to the six variables are

$$\frac{\partial F}{\partial d_{ed1}^{i}}=d_{em}^{i-1}r_{e1}^{i}r_{e1}^{i-1}+d_{em}^{i-1}r_{e1}^{i}r_{e1}^{i}+d_{em}^{i-1}r_{e1}^{i}r_{de1}^{i}-d_{em}^{i}r_{e1}^{i-1}r_{de1}^{i} \tag{61}$$

$$\frac{\partial F}{\partial r_{e1}^{i}}=d_{em}^{i-1}d_{ed1}^{i}(r_{e1}^{i-1}+2r_{e1}^{i}+r_{de1}^{i}) \tag{62}$$

$$\frac{\partial F}{\partial r_{e1}^{i-1}}=d_{em}^{i-1}d_{ed1}^{i}r_{e1}^{i}-d_{em}^{i-1}d_{em}^{i}r_{de1}^{i}-d_{em}^{i}d_{em}^{i}r_{de1}^{i}-d_{ed1}^{i}d_{em}^{i}r_{de1}^{i} \tag{63}$$

$$\frac{\partial F}{\partial r_{de1}^{i}}=d_{em}^{i-1}d_{ed1}^{i}r_{e1}^{i}-d_{em}^{i-1}d_{em}^{i}r_{e1}^{i-1}-d_{em}^{i}d_{em}^{i}r_{e1}^{i-1}-d_{ed1}^{i}d_{em}^{i}r_{e1}^{i-1} \tag{64}$$

$$\frac{\partial F}{\partial d_{em}^{i-1}}=d_{ed1}^{i}r_{e1}^{i}r_{e1}^{i-1}+d_{ed1}^{i}r_{e1}^{i}r_{e1}^{i}+d_{ed1}^{i}r_{e1}^{i}r_{de1}^{i}-d_{em}^{i}r_{de1}^{i}r_{e1}^{i-1} \tag{65}$$

$$\frac{\partial F}{\partial d_{em}^{i}}=-(d_{em}^{i-1}+2d_{em}^{i}+d_{ed1}^{i})r_{e1}^{i-1}r_{de1}^{i} \tag{66}$$

The estimated distance d_{ed1}^{i} is the function of the other variables d_{em}^{i}, d_{em}^{i-1}, r_{e1}^{i}, r_{e1}^{i-1} and r_{de1}^{i}. The difference Δd_{ed1}^{i} is

$$\begin{aligned}\Delta d_{ed1}^{i}&=\frac{\partial d_{ed1}^{i}}{\partial d_{em}^{i}}\Delta d_{em}^{i}+\frac{\partial d_{ed1}^{i}}{\partial d_{em}^{i-1}}\Delta d_{em}^{i-1}+\frac{\partial d_{ed1}^{i}}{\partial r_{e1}^{i}}\Delta r_{e1}^{i}+\frac{\partial d_{ed1}^{i}}{\partial r_{e1}^{i-1}}\Delta r_{e1}^{i-1}+\frac{\partial d_{ed1}^{i}}{\partial r_{de1}^{i}}\Delta r_{de1}^{i}\\&=-(\frac{\partial F}{\partial d_{em}^{i}}\Delta d_{em}^{i}+\frac{\partial F}{\partial d_{em}^{i-1}}\Delta d_{em}^{i-1}+\frac{\partial F}{\partial r_{e1}^{i}}\Delta r_{e1}^{i}+\frac{\partial F}{\partial r_{e1}^{i-1}}\Delta r_{e1}^{i-1}+\frac{\partial F}{\partial r_{de1}^{i}}\Delta r_{de1}^{i})/\frac{\partial F}{\partial d_{ed1}^{i}}\end{aligned} \tag{67}$$

Consider the case that $r_{e1}^{i-1}=r_{e1}^{i}<<r_{de1}^{i}$ and $d_{em}^{i-1}=d_{em}^{i}<<d_{ed1}^{i}$. Applying formulas from (61) to (66)

to equation (67), then Δd_{ed1}^{i} can be approximately expressed as

$$\begin{aligned}\Delta d_{ed1}^{i} \approx & \frac{d_{ed1}^{i}}{(r_{e1}^{i}+r_{e1}^{i-1})r_{e1}^{i}}[(3/d_{ed1}^{i}+1/d_{em}^{i})r_{e1}^{i}r_{de1}^{i}\Delta d_{em}^{i}-(2r_{e1}^{i}/d_{em}^{i}+r_{de1}^{i}/d_{em}^{i}-r_{de1}^{i}/d_{ed1}^{i})r_{e1}^{i}\Delta d_{em}^{i-1} \\ & -(3r_{e1}^{i}+r_{de1}^{i})\Delta r_{e1}^{i}-(r_{e1}^{i}-2d_{em}^{i}r_{de1}^{i}/d_{ed1}^{i}-r_{de1}^{i})\Delta r_{e1}^{i-1}+2(d_{em}^{i}/d_{ed1}^{i})r_{e1}^{i}\Delta r_{de1}^{i}]\end{aligned} \tag{68}$$

If the terms r_{e1}^{i} and $1/d_{ed1}^{i}$ in the numerator of equation (68) is neglected, then equation (68) becomes

$$\frac{\Delta d_{ed1}^{i}}{d_{ed1}^{i}} \approx \frac{r_{de1}^{i}}{(r_{e1}^{i}+r_{e1}^{i-1})}(\frac{\Delta r_{e1}^{i-1}}{r_{e1}^{i-1}}-\frac{\Delta r_{e1}^{i}}{r_{e1}^{i}}) \tag{69}$$

Equation (69) indicates that the main sources of errors in estimating the distance with cross ratio invariance are the errors in r_{e1}^{i} and r_{e1}^{i-1}. The relative error in the estimated distance is proportional to the difference between the relative errors $\Delta r_{e1}^{i-1}/r_{e1}^{i-1}$ and $\Delta r_{e1}^{i}/r_{e1}^{i}$, with a gain of $r_{de1}^{i}/(r_{e1}^{i}+r_{e1}^{i-1})$.

For example, when $r_{e1}^{i-1}=r_{e1}^{i}=5$ and $r_{de1}^{i}=50$ pixels, the relative error $\Delta d_{ed1}^{i}/d_{ed1}^{i}$ is 5 times the error $\Delta r_{e1}^{i-1}/r_{e1}^{i-1}-\Delta r_{e1}^{i}/r_{e1}^{i}$. Therefore, reducing r_{de1}^{i} or increasing r_{e1}^{i} or r_{e1}^{i-1} can reduce the relative error $\Delta d_{ed1}^{i}/d_{ed1}^{i}$.

In general, the distance estimated with cross ratio invariance is still a rough one. When it is used for control purpose, it is better to be combined with an image based visual control method to form an integrated system, or to add a limit unit in the robot controller to confine the step size in the visual.

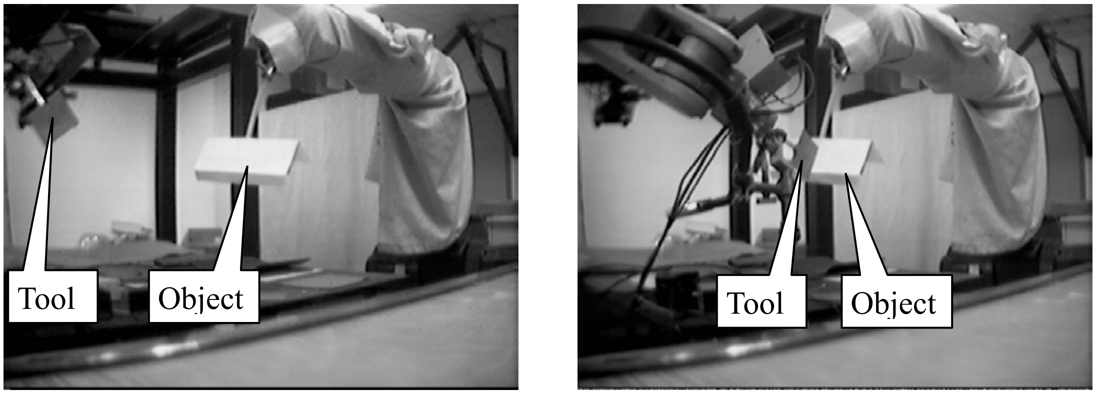
7 Experiment and results

7.1 Experiment 1

The experiment system is shown in Fig.5. The end-effector was mounted with a colored rectangle as the tool. The white rectangle is the object to approach. The end-effector approaches the object's centre along the opposite direction of the surface normal of the planar object. Two cameras were placed near the base of the industrial robot UP6, whose intrinsic parameters M_{in1} and M_{in2} were calibrated using the method proposed in Section 3. The relative pose ${}^1R_2$ is computed with the method in Section 4.

The task was to move the end point of the tool to the centre of the object. In the experiment, the control algorithm adopted was a proportional control law, i.e. with the gains K_i and K_d in formula (46) set to zero. The factor in formula (39) was assigned as k_θ=0.4. The position step factor in formula (42) at the beginning of the approaching was set to k_s=25 mm. The proportional gains were initially determined by simulation and finally tuned empirically in experiments and were set as

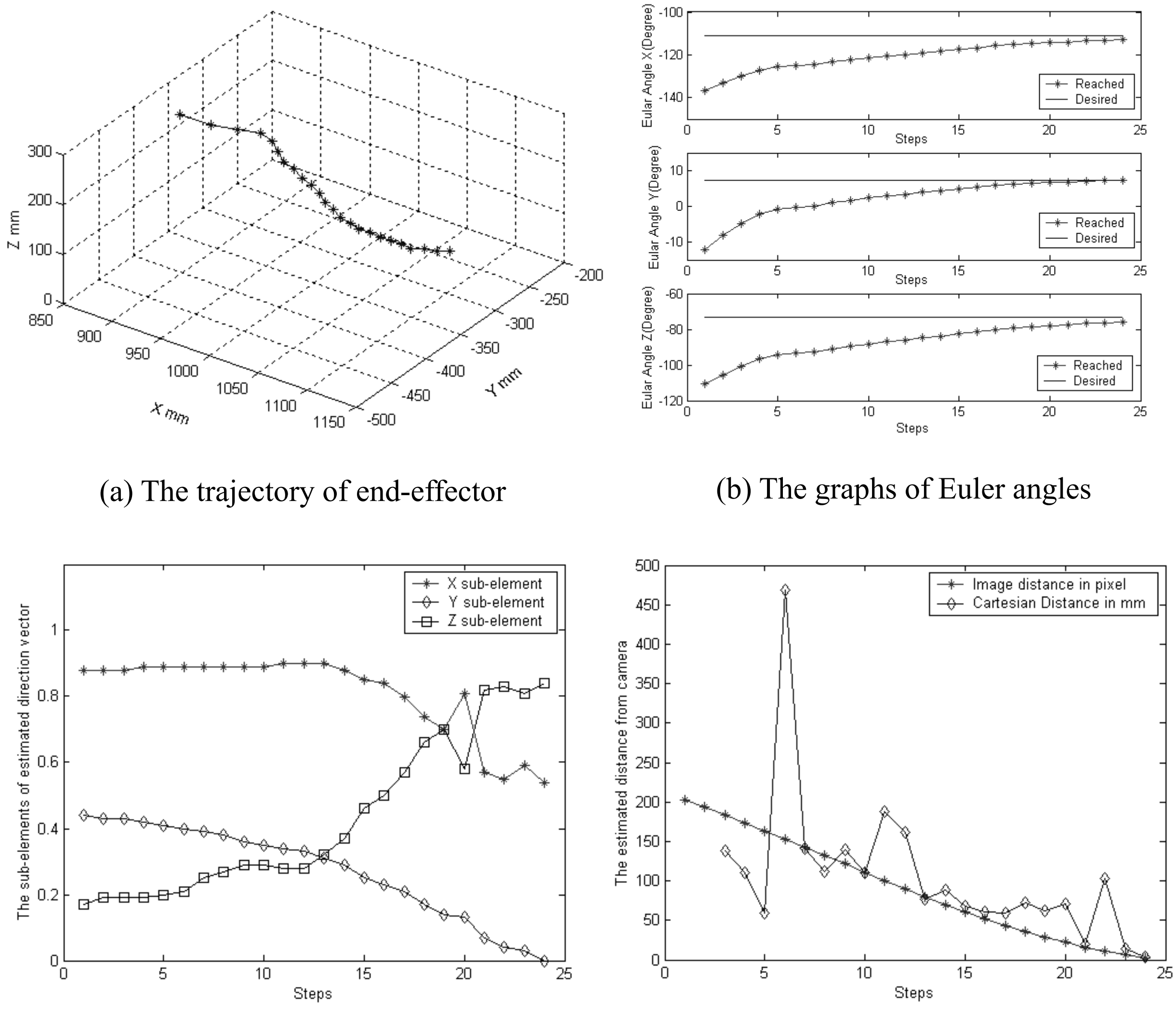
$$K_p = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \tag{70}$$



(a) The scene of experiment at start (b) The scene of experiment in the end

Fig.6 The scenes of experiment

The experiment was conducted using the method in Section 5. Two scenes of the experiment are shown in Fig.6. The scene of the tool and the object, denoted by a color rectangle and a white rectangle respectively, at the start of the experiment is given in Fig.6 (a). Fig.6 (b) shows the scene at the end. It can be seen that the tool reached the desired position and pose well.



(a) The trajectory of end-effector

(b) The graphs of Euler angles

(c) The estimated direction results from the end-effector to object

(d) The distances between the tool and object in Cartesian and image space

Fig.7 The experimental results

Some experimental results are given in Fig.7. Fig.7 (a) shows the trajectory of the end-effector in the experiment. The three steps at the beginning were controlled using a constant step length with formula

(39), (42) and (47). Then the motions of the end-effector were controlled according to formula (39), (46) and (47). The end-effector was moved toward the object. Fig.7 (b) gives the desired and reached orientations of the end-effector in the experiment. It is seen that the orientation of the end-effector changed smoothly and achieved the desired one finally. The components in the estimated result are given in Fig.7 (c). There existed variations in the estimation because of noise disturbance. Fig.7 (d) shows the distances between the tool and the object in Cartesian and image space. The distance in the image space was calculated with the image coordinates of the tool and the object. The distance in Cartesian space was estimated with equation (43) and (44) according to cross ratio invariance. The distance varied considerably due to the sensitivity to noise. Besides the noise, the trajectory of the end-effector, which was not a line, also had a strong influence on the accuracy of the estimated distance. In our experiment, to reduce the aforementioned effect, the estimated distance was filtered with a low-pass filter before it was used to control the end-effector. Also the step size for each visual control sampling was limited to 25mm or less.

7.2 Experiment 2

The purpose of this experiment was to compare our pose-based visual control method with a traditional position-based method. In this experiment, the tool was a bar instead of the rectangle in the last experiment. Different from Section 5.2, the poses here were just the pointing direction of the tool and the normal of the planar object. Assume that the pointing direction of the tool was ${}^{1}\vec{a}_{em}$ in frame

$\mathbb{C}_1$, the normal of the planar object was ${}^1\vec{a}_r$ in frame $\mathbb{C}_1$. The rotation angle ${}^1\theta$ and axis ${}^1\vec{f}$ in frame $\mathbb{C}_1$ were computed via (71). Then the desired pose ${}^1R_{ed}$ was determined with (72). Applying formula (72) to equation (37), the pose adjustment can be determined with formula (38) and (39).

$$\begin{cases} {}^1\vec{f} = {}^1\vec{a}_{em} \times {}^1\vec{a}_r \\ {}^1\theta = arccos({}^1\vec{a}_{em} \cdot {}^1\vec{a}_r) \end{cases} \tag{71}$$

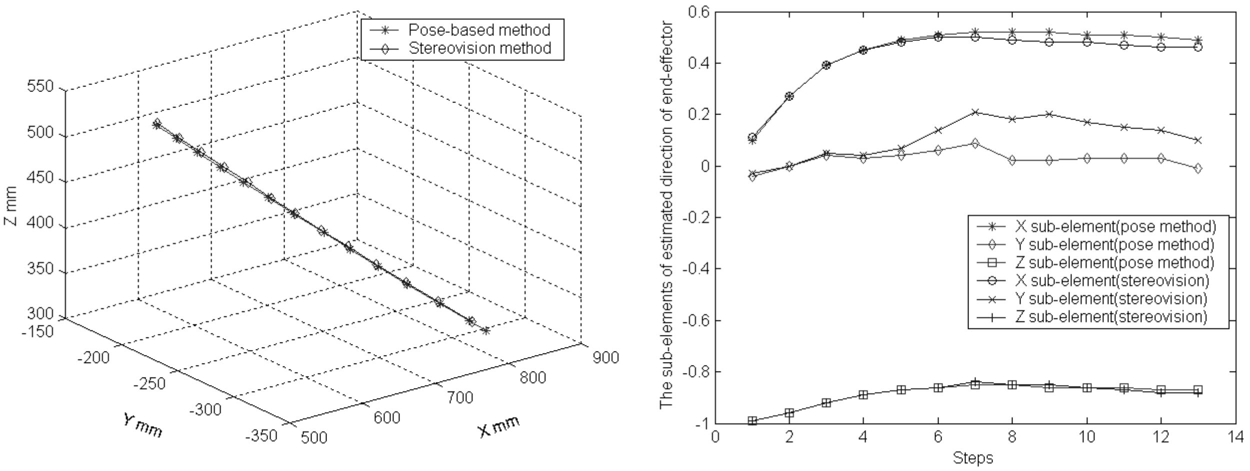
$${}^1R_{ed} = Rot({}^1\vec{f}, {}^1\theta) \tag{72}$$

An experiment for object approaching was conducted with our pose-based visual control method. Similar to the experiment in Section 7.1, the estimated distance was filtered with a low-pass filter before it was used to control the end-effector. The step size for each visual control step was limited to 25mm or less.

Next, the position-based visual control method as described in [2] was used in the comparison experiment. The positions of the tool and target were calculated via stereovision. The poses of the tool and target were obtained via 3D reconstruction [2] in the comparison experiment. The positions and poses of the tool and target were calculated in each control step. The step size for each visual control step was also limited to 25mm or less.

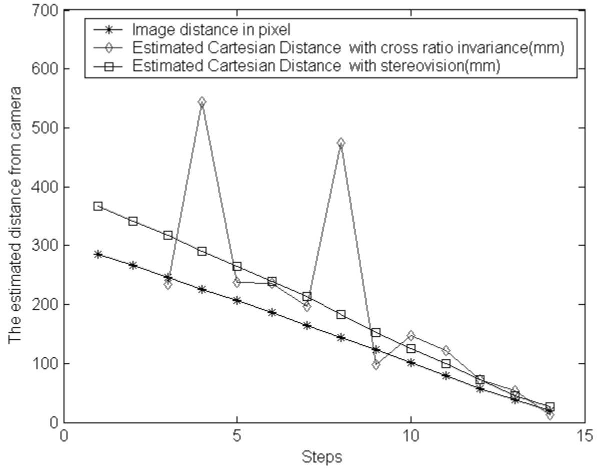
To compare our pose-based method with the position-based one, the results are plotted in Fig.8. The trajectories of the end-effector are shown in Fig.8 (a). Fig.8 (b) displays the actual tool directions in the robot frame. Fig.8 (c) shows the distances between the tool and the object in Cartesian space and image space. It is seen that the estimated distance with cross ratio invariance has a higher precision in the end stage than other stages. This is consistent with the error analysis in Section 5.2. The scenes of experiments in the end stage with pose-based and position-based visual control method are given in Fig.9.

It can be seen from Fig.8 and Fig.9 that the pose-based visual control method has similar effectiveness to the position-based one. However, the pose-based visual control method proposed in this paper is much easier to implement than the position-based one. It avoids the tedious calibration of the extrinsic parameters of cameras and does not need any position information of the environment or metric information of the object.



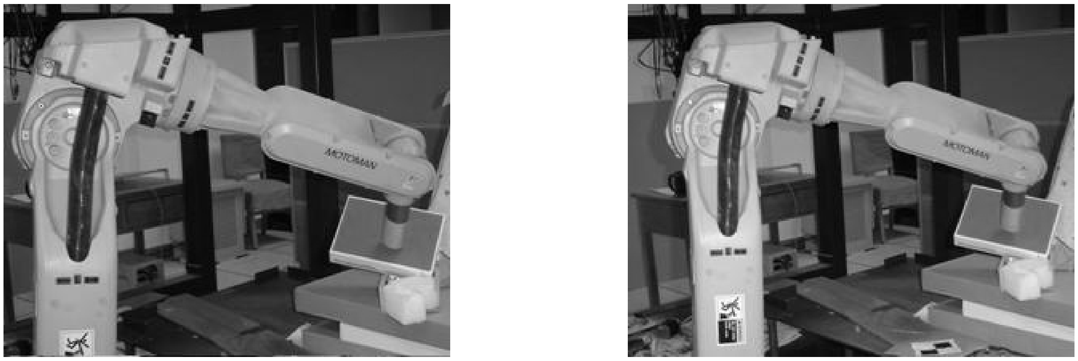
(a) The trajectories of end-effector

(b) The directions of tool



(c) The estimated distances

Fig.8 The results with pose-based and position-based method



(a) The scene with pose-based method

(b) The scene with position-based method

Fig.9 The scenes of experiments in the end stage

It is observed that in some pose detection experiments, the pose of a line could not be obtained with the method in Section 4.2.1. The reason was that the two equations, (32) and (34), were linearly correlated. With the active pose adjustment of the cameras, this problem could be solved as the two equations will then become non-correlated.

8 Conclusions

In this paper, a method for self-calibration of the intrinsic parameters of a camera is investigated using groups of parallel lines, which are orthogonal. If the image coordinates of lens center are known, and there is no difference between the magnification coefficients in the horizontal and vertical direction from the imaging plane coordinates to the image coordinates, then the magnification coefficient k can be self-calibrated with one view of two orthogonal groups of parallel lines. If there are three intrinsic parameters, one view of three groups or three views of two groups of parallel lines in orthogonal are necessary for the intrinsic parameter self-calibration. For a four parameter model, four views of two groups of parallel lines in orthogonal are needed. The analytic solutions for the self-calibration are presented.

With the intrinsic parameters of a camera, the poses of the camera relative to two orthogonal groups of parallel lines in are deduced. For two cameras, their relative pose is computed from their poses relative to the parallel lines. With the intrinsic parameters and transformation between the two cameras, methods are proposed for calculating the poses of a line, plane and rigid object. Compared with stereo vision, our method does not need correspondence matching for feature points which is a main source of errors in 3D reconstruction. Rather line matching is performed and the poses are computed via selecting arbitrary two points on a line. As line matching is more robust than point matching, our method offers a more robust approach for visual control.

A new visual control method is developed using pose detection rather than 3D reconstruction. This avoids the tedious calibration of the extrinsic parameters of cameras and does not need any position information of the environment or metric knowledge about the object. An approaching task is performed with our pose-based visual control to show the application of the method. The object's pose and the orientation of the vector from the tool to the object are estimated with the methods presented in this paper. In approaching the object, the distance between the tool and the object is estimated with cross ratio invariance. The experimental results verified the effectiveness of the proposed methods.

The main features of our methods include automatic calibration of the camera intrinsic parameters based on parallel lines only without requiring metric knowledge of the target or environment, robust determination of the transformation between the two cameras using line matching rather than point matching, and an easily implementable visual control method based on pose detection rather than 3D reconstruction. The developed method will be useful for autonomous systems including mobile manipulators, humanoid robots.

Acknowledgement

The work described in this paper was supported by a grant from the Research Grants Council of Hong Kong [Project No. CityU1206/04E]. The authors would like to thank the National High Technology Research and Development Program of China for supporting this work under grant 2002AA422160. We would also like to thank the National Key Fundamental Research and Development Project of

China (973, No.2002CB312200) for their support.

References

[1] G. D. Hager, S. Hutchinson, and P. I. Corke, A tutorial on visual servo control, IEEE Transaction on Robotics and Automation 12, No. 5, pp651-670, 1996.

[2] Myoungsoo Han, Soongeul Lee, Sung-Kee Park and Munsang Kim, A New Landmark-Based Visual Servoing with Stereo Camera for Door Opening, International Conference on Control, Automation and Systems, 1892-1896, Muju Resort, Jeonbuk, Korea, 2002.

[3] E. Malis, F. Chaumette, and S. Boudet, 2.5D visual servoing, IEEE Transactionon Robotics and Automation, Vol.15, No.2, pp238-250, 1999.

[4] J. Hespanha, Z. Dodds, G. D. Hager, A. S. Morse, What can be down with an uncalibrated stereo system, Proceedings of IEEE International Conference on Robotics and Automation, Vol.2, pp1366-1372, 1998.

[5] D. Kragic, A.T. Miller, P. K. Allen, Real-time tracking meets online grasp planning, Proceedings of IEEE International Conference on Robotics and Automation, Vol.3, pp2460-2465, 2001.

[6] A. Sugimoto, W. Nagatomo, T. Matsuyama, Estimating Ego Motion by Fixation Control of Mounted Active Cameras, Proc. of Asian Conference on Computer Vision, Vol. 1, pp. 67--72, 2004.

[7] Allen Y. Yang, Wei Hong, Yi Ma. Structure and Pose from Single Images of Symmetric Objects with Applications to Robot Navigation, Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taipei, pp1013-1020, 2003.

[8] Takeharu Sato, Jun Sato, Visual servoing from uncalibrated cameras for uncalibrated robots, Systems and Computers in Japan, Vol.31, No.14, pp11-19, 2000.

[9] J. A. Piepmeier, G. V. McMurray, H. Lipkin, Uncalibrated dynamic visual servoing, IEEE Transactions on Robotics and Automation, Vol.20, No.1, pp143-147, 2004

[10] J. A. Piepmeier, G. V. McMurray, H. Lipkin, A Dynamic quasi-Newton method for uncalibrated visual

servoing, Proceedings of IEEE International Conference on Robotics and Automation, Vol.2, pp1595-1600, 1999

[11] J. A. Piepmeier, G. V. McMurray, H. Lipkin, A Dynamic Jacobian estimation method for uncalibrated visual servoing, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp944-949, 1999

[12] J. A. Piepmeier, Harvey Lipkin, Uncalibrated eye-in-hand visual servoing, International Journal of Robotics Research, Vol.22, No.10-11, pp805-819, 2003.

[13] Jiang Qian, Jianbo Su, Online estimation of image Jacobian Matrix by Kalman-Bucy filter for uncalibrated stereo vision feedback, Proceedings of IEEE International Conference on Robotics and Automation, Vol.1, pp562-567, 2002.

[14] Jian-Bo Su, Wen-Bin Qiu, Robotic calibration-free hand-eye coordination based on auto disturbances rejection controller, Acta Automatica Sinica, Vol.29, No.2, pp161-167, 2003

[15] Florian Schramm, Guillaume Morel, Alain Micaelil, Anne Lottin, Extended-2D visual servoing, Proceedings of 2004 IEEE International Conference on Robotics and Automation, pp267-273, 2004

[16] P. C. P. Carvalho, F. Szenberg, M. Gattass, Image-based modeling using a two-step camera calibration method. Proceedings of International Symposium on Computer Graphics, Image Processing and Vision, pp388-395, 1998.

[17] M. Bénallal, J. Meunier, Camera calibration with simple geometry. The 2003 International Conference on Image and Signal Processing, 2003.

[18] E. Guillou, D. Meneveaux, E. Maisel, K. Bouatouch, Using vanishing points for camera calibration and coarse 3D reconstruction from a single image, Visual Computer, v 16, n 7, pp396-410, 2000.

[19] D. H. Lee, K. H. Jang, S. K. Jung, Intrinsic camera calibration based on radical center estimation, Proceedings of the International Conference on Imaging Science, Systems and Technology, pp7-13, 2004.

[20] J. Kosecka, W. Zhang, Efficient computation of vanishing points, Proceedings of IEEE International Conference on Robotics and Automation, Washington, DC, United States, Vol.1, pp223-228, 2002

[21] A. Almansa, A. Desolneux, Vanishing point detection without any a priori information, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.25, No.4, pp502-507, 2003.

[22] O. D. Faugeras, G. Toscani, The calibration problem for stereo. IEEE Compuyer Society Conference on Computer Vision and Pattern Recognition, Minmi Beach, Florida, 1986.

[23] S. D. Ma, “A self-calibration technique for active vision system,” IEEE Transaction on Robotics and Automation, Vol.12, No.1, pp114-120, 1996.

[24] D. Xu, Y. F. Li and M. Tan, “A Method for Calibrating Cameras with Large Distortions in Lens”, *Optical Engineering,* 2006 (in press).

Biographies:

De Xu graduated from Shandong University of Technology (SUT), China, in 1985. He received the Master degree from SUT in 1990 and the Ph.D. degree from Zhejiang University, China, in 2001. He has been with Institute of Automation, Chinese Academy of Sciences (IACAS) since 2001. He is a professor with the Laboratory of Complex Systems and Intelligence Science, IACAS. He is a senior researcher associate with City University of Hong Kong. His research interests include robotics and automation, especially the control of robots such as visual control and intelligent control.

You Fu Li obtained his BSc and MSc degrees in Electrical Engineering from Harbin Institute of Technology (HIT), China in 1982 and 1986 respectively. He obtained his PhD degree in robotics from the

Department of Engineering Science, Oxford University, UK, in 1993. From 1993 to 1995, he worked as a post-doctoral researcher in the Department of Computer Science at the University of Wales, Aberystwyth, UK. He joined City University of Hong Kong in 1995. Dr. Li has served as program committee member of different international conferences. His research interests include robot vision, visual tracking, robot sensing and sensor based control, mechatronics and automation. In these areas, he has published over a hundred papers in refereed international journals and conferences. He is an associate editor of ***IEEE*** *Transactions on Automation Science and Engineering* (T-ASE).

Yang Shen graduated from University of Science and Technology of China (USTC), Hefei, China, in 2002. He is a Ph.D. candidate with IACAS. His research interests include robotics and automation.

Min Tan graduated from Tsing Hua University, China, in 1986. He received the Ph.D. degree in 1990 from IACAS. He is a professor with the Laboratory of Complex Systems and Intelligence Science, IACAS. He has published over 100 papers in journals, books and conferences. His research interests include robotics and complex system theory.