Robotica (2017) volume 35, pp. 2218–2237. © Cambridge University Press 2017
doi:10.1017/S0263574716000849

# Multi-cameras visual servoing for dual-arm coordinated manipulation

Jiadi Qu, Fuhai Zhang*, Yili Fu* and Shuxiang Guo

*State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, P. R. China.*
*E-mails: qujiadi@hotmail.com, 18745007416@163.com*

(Accepted December 20, 2016. First published online: January 12, 2017)

**SUMMARY**

Although image-based visual servoing (IBVS) provides good performance in many dual-arm manipulation applications, it reveals some fatal limitations when dealing with a large position and orientation uncertainty. The object features may leave the camera's field of view, and the dual-arm robot may not converge to their goal configurations. In this paper, a novel vision-based control strategy is presented to resolve these limitations. A visual path planning method for dual-arm end-effector features is proposed to regulate the large initial poses to the pre-alignment poses. Then, the visual constraints between the position and orientation of two objects are established, and the sequenced subtasks are performed to attain the pose alignment of two objects by using a multi-tasks IBVS method. The proposed strategy has been implemented on a MOTOMAN robot to perform the alignment tasks of plug–socket and cup–lid, and results indicate that the plug and socket with the large initial pose errors 145.4 mm, 43.8° (the average errors of three axes) are successfully aligned with the allowed pose alignment errors 3.1 mm, 1.1°, and the cup and lid with the large initial pose errors 131.7 mm, 20.4° are aligned with the allowed pose alignment errors −2.7 mm, −0.8°.

KEYWORDS: Dual-arm, Visual servoing, Visual path planning, Visual constraints, Uncertainty.

**Nomenclature**

| | |
|---|---|
| $A$ | Active matrix |
| $\boldsymbol{e}$ | The error of the image feature |
| $H$ | The cost function of joint-limit avoidance |
| $\boldsymbol{J}$ | Jacobian matrix |
| $\boldsymbol{K}$ | Camera intrinsic matrix |
| $\boldsymbol{L}$ | Interaction matrix of the feature point |
| $\boldsymbol{q}, \dot{\boldsymbol{q}}$ | Joint position and joint velocity |
| $R, t$ | Rotation matrix, transformation matrix |
| $\boldsymbol{s}$ | Image feature coordinates |
| $\boldsymbol{v}$ | Velocity screw |
| $\sigma$ | Time instant |
| $\rho, \theta$ | Polar parameters of line |
| $\lambda$ | Control gain |
| $\Sigma$ | Coordinate frame |

**Subscripts and superscripts:**

| | |
|---|---|
| $B$ | Robot base |
| $c$ | Object's centroid |
| $L, R$ | Left arm, right arm |
| $M, F$ | Mobile camera, fixed camera |
| $n$ | The feature of the tip point |
| $T$ | The feature of the terminal point |

* Corresponding author. E-mail: zfhhit@hit.edu.cn, meylfu_hit@163.com

| | |
|---|---|
| $p, h$ | Peg, hole |
| $s$ | Task |
| 0,1 | Initial state, desired state |
| $*$ | Desired |

## 1. Introduction

Dual-arm manipulation is currently widely applied in both domestic and industrial settings. Examples include the carrying of heavy loads,[1] the manipulation of flexible objects,[2,3] the assembly of machine parts,[4] the assistance of persons with disabilities,[5] and the automation of common kitchen tasks.[6] Coordination tasks can be divided into two categories: coordinated symmetric (two-arm manipulation of one object) and coordinated asymmetric (two-arm manipulation of different objects).[7] To perform complex manipulation tasks, dual-arm robot systems need to have enhanced sensing capabilities. These capabilities are attained through the use of exteroceptive sensors like force and vision sensors.[8]

In a dual-arm robotic platform, visual systems usually consist of two or more cameras that can either be rigidly attached to the robot's end-effectors (mobile camera), or in a fixed location observing the robot's workspace (fixed camera).[9] The configuration of multiple cameras allows the robot to receive input from complementary sources of information during task completion.[8,10] In Ref. [11], the concept of multi-cameras visual servoing is proposed, describing robot controlled visual servoing using several cameras. So far, dual-arm visual servoing techniques are simply extensions of those used for single-arm servoing. Therefore, they are classified under the same two categories as single-arm techniques: position-based (3D visual servoing) and image-based (2D visual servoing).[12,13]

In Ref. [14], 3D visual servoing with a hybrid fixed/mobile camera system is applied to dual-arm manipulation based on the real-time estimation of a target object's pose using the extended Kalman filter. In Ref. [15], a dual-arm robot manipulation is presented in which a nut and bolt are screwed together. The features of the nut and bolt are recognized by a stereo-fixed camera and two mobile cameras, the alignment of the nut and bolt is interpreted using 3D visual servoing. Similarly, several manipulations in a domestic setting are demonstrated by using the humanoid robot Armar-? equipped with a multi-cameras system.[16,17] In general, 3D visual servoing relies on the calibration accuracy of camera and object models.[18]

In addition, due to the good local properties of convergence and stability, as well as the toleration for calibration error, 2D visual servoing is more applicable than 3D visual servoing when applied to robotic dual-arm manipulation. In Ref. [19], 2D visual servoing with binocular stereo vision using the SCARA type dual-arm robot is applied to an assembly task, where the exact image Jacobian is calculated to control the position and orientation of a robot with respect to the manipulated object. Moreover, dual-arm robots, utilizing 2D visual servoing based on the features from two fixed cameras, can be used in minimally invasive surgeries.[20] The position and orientation of surgical instruments are manipulated to tie surgical knots. In Ref. [21], 2D features of objects are observed in real time by two mobile cameras, and two arms are controlled simultaneously to grasp a single large object. The Eurobot Ground Prototype in Ref. [22] performs similar object grasping and capturing through the use of a dual-arm space manipulator.[23] In Ref. [24], hybrid fixed/mobile camera systems are used to observe the alignment of a pen and pen cap, two arms are controlled by 2D visual servoing to place the cap back on the pen.

Asymmetric coordination tasks aim to achieve a specific pose alignment (position and orientation) between two objects gripped by the robot's end-effectors (hereinafter referred to as two objects).[25] As far as we know, the current works that deal with dual-arm manipulation using multi-cameras visual servoing, especially 2D visual servoing, only consider the alignment of two objects with small initial pose errors. However, a real dual-arm manipulation task is usually executed on objects with large initial relative pose errors. Existing methods' lack of local stability can lead to a robot completing its task in a manner which is not optimal. This lack of local stability can also lead to a robot attempting to complete a physically invalid task or a task which exceeds its joint limits or field-of-view limits; in these cases, the errors can lead to task failure.[26,27] Furthermore, large initial pose errors between two objects can also lead to the features of the object's pose not even being observed or only partly observed by the mobile camera, also leading to the inability to complete the given task. To adopt the existing methods, the initial pose error between the two objects should be limited to the appropriate

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

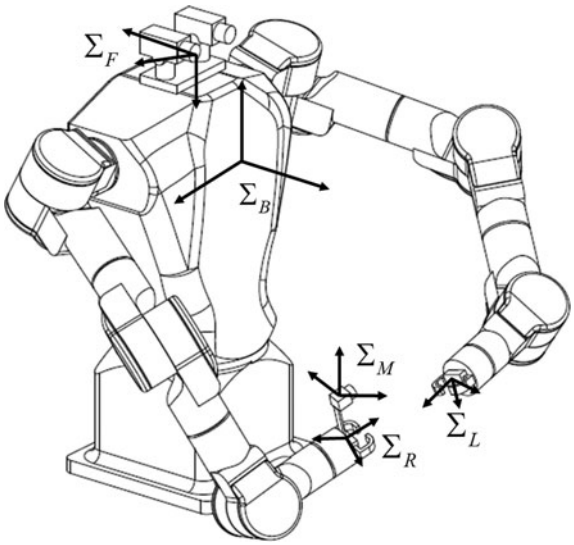



Fig. 1. Dual-arm robot model and coordinate frames.

range. Therefore, only two objects with small initial pose errors can be controlled, and only after "teaching-by-showing" in advance. This fact shows the strong lack of flexibility in these methods.

In this paper, the image-based visual servoing control is developed for a large-scale application of dual-arm manipulation. To solve the large initial pose uncertainty of two objects, this paper proposed a novel multi-cameras visual servoing control strategy for the achievement of the pose alignment task, which combines the visual path planning with multi-tasks IBVS method. A visual path planning method is proposed to deal with large initial position and orientation uncertainty of dual-arm robot. Based on the projective model and dual-arm coordination constraints, the two desired image trajectories of two effector features in the fixed camera plane are generated synchronously. The proposed method extends IBVS to perform a large-scale dual-arm manipulation task. A multi-tasks IBVS method is developed for the position and orientation control of robot manipulation. The pose features extracted from the general peg-and-hole are established both in the fixed camera and in the mobile camera, and the sequenced subtasks are performed for the pose alignment by using the derived multi-tasks image Jacobian matrix.

Concretely, the alignment task is divided into two steps:

1. Within the fixed camera space, the robot's two end-effectors are controlled through 2D visual servoing to pre-alignment the two objects. Due to the large initial pose errors, a visual path planning method has been developed to generate the desired trajectories of the two end-effectors. This ensures the local convergence and stability of the 2D visual servoing.
2. The two objects are controlled through 2D visual servoing to move from their pre-alignment poses to the final alignment. This alignment is achieved through the completion of four separate visual alignment tasks. An active matrix was introduced to ensure these tasks are completed sequentially.

## 2. Dual-Arm Robot and Task Modeling

### *2.1. Robot model*

The robotic platform MOTOMAN SDA5F with two 7-DOF robotic arms is used in this paper. Two cameras are equipped to perform the required visual tasks. A stereo camera, called the fixed camera, is fixed on the head to give a global view. In this work, only one stereo-type camera is used. A monocular camera is located on the robot's right forearm called the mobile camera. These two cameras together provide a visual information stream of the scene. This visual information stream is used to complete the dual-arm coordinated manipulation. As shown in Fig. 1, several frames are introduced.

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

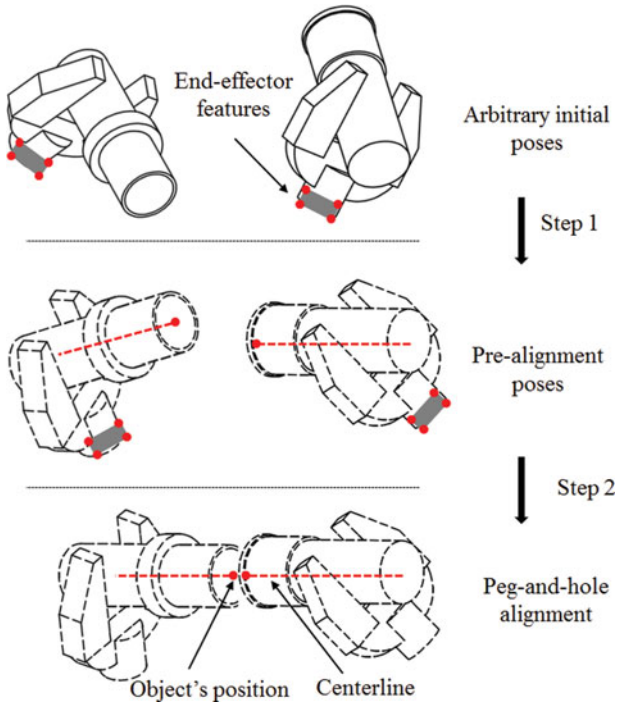



Fig. 2. Two steps for the pose alignment of the peg and hole.

### *2.2. Task model*

In this paper, a general peg-and-hole alignment task is performed by a dual-arm robot to verify the proposed strategy. It should be noted that the general peg and hole represents cylindrical multi peg–holes and cube peg–hole, such as plug and socket and cup and lid, etc. First, the following explanations are given:

1. The peg and hole are mounted on the left- and right end-effector, respectively.
2. Under the large initial pose errors, both the features fixed on the two end-effectors can be observed by the fixed camera at all times.
3. The pose alignment of the peg and hole are considered that the orientation of the centerlines and the position of the terminal points need to keep consistent.
4. The accuracy of the pose alignment is defined: The position error of each axis < 5 mm, and the orientation error of each direction < 2° (it is determined by teaching-by-showing in advance).

To realize the relative pose alignment of two objects with large initial pose errors, the task is divided into two steps as shown in Fig. 2:

Step 1: Move the two end-effectors from the initial poses to pre-alignment poses by observing the features of the two end-effectors in the fixed camera space, which consists of four mark points.

Step 2: Move the plug from the pre-alignment pose to the desired pose by observing the object features both in the fixed and mobile camera spaces, which consist of the centerline, the position of the tip point, and the mark point.

Note: the pre-alignment poses are obtained offline, which can be the optimal poses in terms of robot workspace and camera resolution.

## 3. Visual Path Planning for Dual-Arm End-Effector Features

Visual path planning method is effective to extend IBVS to a large-scale application. The projective space model based visual path planning method is presented in Ref. [28]. The image trajectories can be obtained by associating the planned projective path with the initial and finished image features.

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

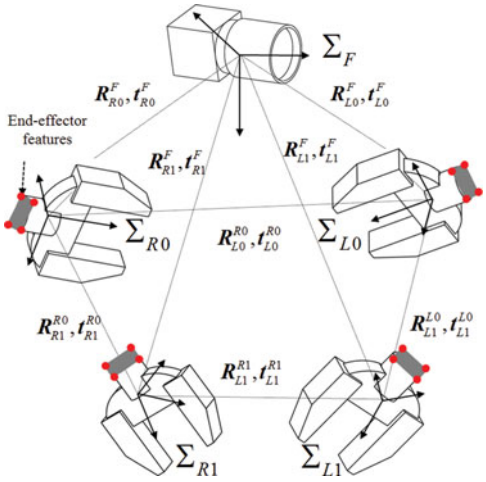



Fig. 3. The relationships of different frames at the initial and desired poses of the two end-effectors.

The presented method is applied to a single-arm application. In this work, we focus on the dual-arm manipulation. If the above method is respectively applied to the two arms (dual-arm robot are considered as two single-arm robots), the pose synchrony between two arms cannot be ensured. Therefore, the dual-arm coordination constraints need to be introduced to the projective space model. The two desired image trajectories of two end-effectors are generated synchronously, only through the planned projective trajectories of one end-effector.

The large initial pose errors can bring about the object's features being not even observed or only partly observed by the cameras. Therefore, the end-effector features are selected for ensuring the visibility of the image features during the task process.

As shown in Fig. 3, the relationships between the coordinate frames of the fixed camera $\Sigma_F$ and two end-effectors $\Sigma_L$, $\Sigma_R$ are established, and "0" and "1" denote the initial state and the finished state of two end-effectors. The two end-effector features are observed by the fixed camera, and the camera pin-hole model are used: $z_i[\boldsymbol{s}_i \ 1]^T = \boldsymbol{K}x_i^F$, which indicate the mapping from the image coordinate $\boldsymbol{s}_i$ to the Cartesian coordinate $\boldsymbol{x}_i^F$ of a feature point $i$ by using the depth $z_i$ and the camera intrinsic matrix $\boldsymbol{K}$. The Cartesian coordinates, $\boldsymbol{x}_i^{R0}$ in $\Sigma_{R0}$ and $\boldsymbol{x}_i^{R1}$ in $\Sigma_{R1}$, can be obtained by the coordination transformations $(\boldsymbol{R}^F_{R0}, \boldsymbol{t}^F_{R0})$ and $(\boldsymbol{R}^F_{R1}, \boldsymbol{t}^F_{R1})$. Thus, the projective model of the feature point $i$ on the right end-effector are obtained,

$$z_i^{R1}\begin{bmatrix}\boldsymbol{s}_i^{R1} & 1\end{bmatrix}^T = \boldsymbol{G}^{R1}_{R0}\left(z_i^{R0}\begin{bmatrix}\boldsymbol{s}_i^{R0} & 1\end{bmatrix}^T - \boldsymbol{g}_{R0}\right) + \boldsymbol{g}_{R0} + \boldsymbol{g}^{R1}_{R0} \tag{1}$$

With the projective parameters: $\boldsymbol{G}^{R1}_{R0} = \boldsymbol{K}\boldsymbol{R}^F_{R0}\boldsymbol{R}^{R0}_{R1}(\boldsymbol{R}^F_{R0})^T\boldsymbol{K}^{-1}$, $\boldsymbol{g}_{R0} = \boldsymbol{K}\boldsymbol{t}^F_{R0}$ and $\boldsymbol{g}^{R1}_{R0} = \boldsymbol{K}\boldsymbol{R}^F_{R0}\boldsymbol{t}^{R0}_{R1}$. Accordingly, the projective model of the feature point $i'$ on the left end-effector can be given as,

$$z_{i'}^{L1}\begin{bmatrix}\boldsymbol{s}_{i'}^{L1} & 1\end{bmatrix}^T = \boldsymbol{G}^{L1}_{L0}\left(z_{i'}^{L0}\begin{bmatrix}\boldsymbol{s}_{i'}^{L0} & 1\end{bmatrix}^T - \boldsymbol{g}_{L0}\right) + \boldsymbol{g}_{L0} + \boldsymbol{g}^{L1}_{L0} \tag{2}$$

with $\boldsymbol{G}^{L1}_{L0} = \boldsymbol{K}\boldsymbol{R}^F_{L0}\boldsymbol{R}^{L0}_{L1}(\boldsymbol{R}^F_{L0})^T\boldsymbol{K}^{-1}$, $\boldsymbol{g}_{L0} = \boldsymbol{K}\boldsymbol{t}^F_{L0}$, $\boldsymbol{g}^{L1}_{L0} = \boldsymbol{K}\boldsymbol{R}^F_{L0}\boldsymbol{t}^{L0}_{L1}$.

The asymmetric coordination manipulation needs each arm to perform their objective task, while it also needs to consider the coordination constraints of time synchronization and relative pose. Therefore, the desired trajectories of one end-effector feature should be derived from those of the other end-effector. Furthermore, $\boldsymbol{G}^{L1}_{L0}$, $\boldsymbol{g}_{L0}$, and $\boldsymbol{g}^{L1}_{L0}$ should be derived by $\boldsymbol{G}^{R1}_{R0}$, $\boldsymbol{g}_{R0}$, and $\boldsymbol{g}^{R1}_{R0}$.

The dual-arm coordination constraints $\boldsymbol{R}^R_L$, $\boldsymbol{t}^R_L$ are introduced to the projective model (1) and (2). Furthermore, the projective parameters of left end-effectors can be derived by the ones of right

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

end-effectors,

$$\begin{cases} \boldsymbol{G}_{L0}^{L1} = \boldsymbol{G}_{R0}^{R1}\boldsymbol{G}_{R}^{L} \\ \boldsymbol{g}_{L0} = \boldsymbol{g}_{R0} + \boldsymbol{g}_{R0}^{L0} \\ \boldsymbol{g}_{L0}^{L1} = \boldsymbol{g}_{R0}^{R1} + \boldsymbol{g}_{R1}^{L1} - \boldsymbol{g}_{R0}^{L0} \end{cases} \tag{3}$$

where $\boldsymbol{G}_{R}^{L} = \boldsymbol{K}\boldsymbol{R}_{L0}^{F}\boldsymbol{R}_{R0}^{L0}\boldsymbol{R}_{L1}^{R1}(\boldsymbol{R}_{L0}^{F})^{T}\boldsymbol{K}^{-1}$, $\boldsymbol{g}_{R0}^{L0} = \boldsymbol{K}\boldsymbol{R}_{R0}^{F}\boldsymbol{t}_{L0}^{R0}$, and $\boldsymbol{g}_{R1}^{L1} = \boldsymbol{K}\boldsymbol{R}_{R1}^{F}\boldsymbol{t}_{L1}^{R1}$. $\boldsymbol{G}_{R}^{L}$, $\boldsymbol{g}_{R0}^{L0}$ and $\boldsymbol{g}_{R1}^{L1}$ can be defined as the relative projective parameters of the two end-effector features. Substituting Eq. (3) into Eq. (2), we can obtain,

$$z_{i'}^{L1}\left[\boldsymbol{s}_{i'}^{L1}\ 1\right]^{T} = \boldsymbol{G}_{R0}^{R1}\boldsymbol{G}_{R}^{L}\left(z_{i'}^{L0}\left[\boldsymbol{s}_{i'}^{L0}\ 1\right]^{T} - \boldsymbol{g}_{R0} - \boldsymbol{g}_{R0}^{L0}\right) + \boldsymbol{g}_{R0} + \boldsymbol{g}_{R0}^{R1} + \boldsymbol{g}_{R1}^{L1} \tag{4}$$

where $\boldsymbol{G}_{R}^{L}$, $\boldsymbol{g}_{R0}^{L0}$, $\boldsymbol{g}_{R1}^{L1}$, and $\boldsymbol{g}_{R0}$ are constants, which are related to intrinsic and extrinsic camera parameters, the initial and desired pose of the two end-effectors. In order to obtain the time-varying parameters, the interpolations of $\boldsymbol{G}_{R0}^{R1}$, $\boldsymbol{g}_{R0}^{R1}$ are required.

At instant $\sigma$, the desired trajectories of the feature points $i$ and $i'$ are denoted as follows:

$$\begin{cases} z_{i}^{R\sigma}\left[\boldsymbol{s}_{i}^{R\sigma}\ 1\right]^{T} = \boldsymbol{G}_{R0}^{R\sigma}\left(z_{i}^{R0}\left[\boldsymbol{s}_{i}^{R0}\ 1\right]^{T} - \boldsymbol{g}_{R0}\right) + \boldsymbol{g}_{R0} + \boldsymbol{g}_{R0}^{R\sigma} \\ z_{i'}^{L\sigma}\left[\boldsymbol{s}_{i'}^{L\sigma}\ 1\right]^{T} = \boldsymbol{G}_{R0}^{R\sigma}\boldsymbol{G}_{R}^{L}\left(z_{i'}^{L0}\left[\boldsymbol{s}_{i'}^{L0}\ 1\right]^{T} - \boldsymbol{g}_{R0} - \boldsymbol{g}_{R0}^{L0}\right) + \boldsymbol{g}_{R0} + \boldsymbol{g}_{R0}^{R\sigma} + \boldsymbol{g}_{R1}^{L1} \end{cases} \tag{5}$$

where the boundary conditions $\sigma = 0$ and $\sigma = 1$ are given,

$$\begin{aligned} &\boldsymbol{G}_{R0}^{R\sigma}|_{\sigma=0} = \boldsymbol{I}_{3},\ \boldsymbol{g}_{R\sigma}|_{\sigma=0} = \boldsymbol{g}_{R0},\ \boldsymbol{g}_{R0}^{R\sigma}|_{\sigma=0} = 0; \\ &\boldsymbol{G}_{R0}^{R\sigma}|_{\sigma=1} = \boldsymbol{G}_{R0}^{R1},\ \boldsymbol{g}_{R\sigma}|_{\sigma=1} = \boldsymbol{g}_{R1},\ \boldsymbol{g}_{R0}^{R\sigma}|_{\sigma=1} = \boldsymbol{g}_{R0}^{R1}; \end{aligned}$$

where $\boldsymbol{g}_{R1} = \boldsymbol{g}_{R0} + \boldsymbol{g}_{R0}^{R1}$. The parameters $\boldsymbol{g}_{R0}^{R\sigma}$ and $\boldsymbol{G}_{R0}^{R\sigma}$ are related to the time variable of $\sigma$, and the others are constant in Eq. (5). First, linear interpolation will be used for obtaining $\boldsymbol{g}_{R0}^{R\sigma}$ as follows:

$$\boldsymbol{g}_{R0}^{R\sigma} = \sigma\boldsymbol{g}_{R0}^{R1} \quad \sigma \in [0, 1] \tag{6}$$

In order to obtain $\boldsymbol{G}_{R0}^{R\sigma}$, it needs to the interpolation of $\boldsymbol{G}_{R0}^{R1} = \boldsymbol{K}\boldsymbol{R}_{R0}^{F}\boldsymbol{R}_{R1}^{R0}(\boldsymbol{R}_{R0}^{F})^{T}\boldsymbol{K}^{-1}$, where $\boldsymbol{K}$ and $\boldsymbol{R}_{R0}^{F}$ are constant, and the interpolation of $\boldsymbol{G}_{R0}^{R1}$ can be considered as the interpolation of $\boldsymbol{R}_{R1}^{R0}$. In Ref. [29], $\boldsymbol{R}_{R1}^{R0}$ can be expressed via eigen decomposition, $\boldsymbol{R}_{R1}^{R0} = \boldsymbol{U}\boldsymbol{\Lambda}(\beta)\boldsymbol{U}^{*}$, where the rotation angle $\beta$ in the matrix $\boldsymbol{\Lambda}$ of eigenvalues and rotation axes $u$ in the matrix $\boldsymbol{U}$. Thus, the interpolation of $\boldsymbol{G}_{R0}^{R1}$ is,

$$\boldsymbol{G}_{R0}^{R\sigma} = \boldsymbol{V}\boldsymbol{\Lambda}(\sigma\beta)\boldsymbol{V}^{-1} \quad \sigma \in [0, 1] \tag{7}$$

where $\boldsymbol{V} = \boldsymbol{K}\boldsymbol{R}_{R0}^{F}\boldsymbol{U}$. Hence, the desired trajectories of the feature point $i$ and $i'$ are given,

$$\begin{cases} z_{i}^{R\sigma}\left[\boldsymbol{s}_{i}^{R\sigma}\ 1\right]^{T} = \boldsymbol{G}_{R0}^{R\sigma}\left(z_{i}^{R0}\left[\boldsymbol{s}_{i}^{R0}\ 1\right]^{T} - \boldsymbol{g}_{R0}\right) + \boldsymbol{g}_{R0} + \boldsymbol{g}_{R0}^{R\sigma}, \sigma \in [0, 1] \\ z_{i'}^{L\sigma}\left[\boldsymbol{s}_{i'}^{L\sigma}\ 1\right]^{T} = \boldsymbol{G}_{R0}^{R\sigma}\boldsymbol{G}_{R}^{L}\left(z_{i'}^{L0}\left[\boldsymbol{s}_{i'}^{L0}\ 1\right]^{T} - \boldsymbol{g}_{R0} - \boldsymbol{g}_{R0}^{L0}\right) + \boldsymbol{g}_{R0} + \boldsymbol{g}_{R0}^{R\sigma} + \boldsymbol{g}_{R1}^{L1}, \sigma \in [0, 1] \end{cases} \tag{8}$$

The image coordinates of all $j$ feature points are given by,

$$\boldsymbol{s}_{L}(t) = \left(\left(\boldsymbol{s}_{L1}(t)\right)^{T}, \left(\boldsymbol{s}_{L2}(t)\right)^{T}, \cdots, \left(\boldsymbol{s}_{Lj}(t)\right)^{T}\right)^{T},$$

$$\boldsymbol{s}_{R}(t) = \left(\left(\boldsymbol{s}_{R1}(t)\right)^{T}, \left(\boldsymbol{s}_{R2}(t)\right)^{T}, \cdots, \left(\boldsymbol{s}_{Rj}(t)\right)^{T}\right)^{T}.$$

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

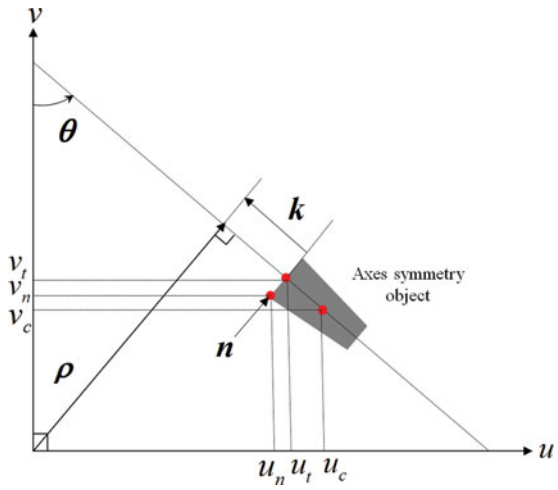



Fig. 4. Visual features between the peg and hole.

In Section 5, the above trajectories will be used as the desired trajectories for 2D visual servoing, and will be defined using the following notation of $s_L^*(t)$, $s_R^*(t)$.

## 4. Visual Constraints of the Object Pose Features

In order to implement the position and orientation alignment of peg and hole, the features of the centerlines and terminal point will be extracted, which is suited to the position and orientation control of robot manipulation. In Ref. [24], the centerline and the terminal point of cylindrical single peg–hole are extracted to perform the alignment task. However, the only two directions, the heading angle and the pitch angle, had been discussed, which cannot deal with a general peg-and-hole task (e.g., cylindrical multi peg–holes, cube peg–hole). Therefore, the image feature of the rotational direction needs to be introduced. Furthermore, a new multi-tasks image Jacobian will be derived in this paper.

### *4.1. Visual features of the objects*

The centerline of the object (grey area) can be represented as shown in Fig. 4. The line features can be denoted by the polar parameters $(\rho, \theta)$, where $\rho$ is the distance from the centerline to the origin of the image plane, and $\theta$ is the orientation angle of the centerline with respect to the vertical axis. The expressions can be shown as follows:

$$\theta = \frac{1}{2}\arctan\left(\frac{2I_{xy}}{I_x - I_y}\right) \tag{9}$$

$$\rho = u_c \cdot \cos(\theta) + v_c \cdot \sin(\theta) \tag{10}$$

In this paper, the centerline of peg and hole are detected and localized by using Hough transform. The image feature $k$ of the terminal point (Fig. 4) is given,

$$k = u_t \cdot \cos(\theta) - v_t \cdot \sin(\theta) \tag{11}$$

In terms of many peg-and-hole tasks, the relative rotational direction needs to be considered. Therefore, the tip point on the object surface recognizes the rotational direction. The image coordinate of the tip point is defined as,

$$n = (u_n, v_n)^T \tag{12}$$

In addition, the image feature $(u_n, v_n)$ can be directly measured by the cameras, whereas the image feature $(u_t, v_t)$ cannot be observed directly owing to the shooting angle of the two cameras. Therefore,

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

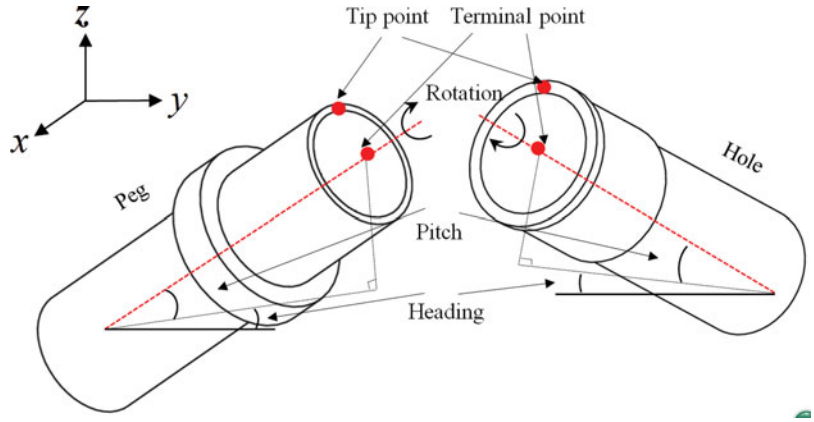



Fig. 5. Position and orientation alignment of the peg and hole.

it needs to establish the relation between $(u_n, v_n)$ and $(u_t, v_t)$ as follows:

$$\begin{cases} u_t = u_n + l/2 \cdot \sin(\theta) \\ v_t = v_n + l/2 \cdot \cos(\theta) \end{cases} \tag{13}$$

where $l$ is the projection on the image plane of the diameter in the end plane.

*4.2. Visual constraints between two objects*

The orientation of the object can be represented by the heading angle, the pitch angle, and the rotation angle, and the position can be represented by the coordinate of the terminal point (Fig. 5). The aim of the alignment is to make the errors of the each orientation and the coordinates of the terminal point between two objects converge to zero. Therefore, the visual constraints between the relative pose of the two objects, both in the fixed and mobile image planes need to be established.

1. Co-planar constraint: The alignment of heading angle

To realize the heading angle alignment of the peg and hole, two planes between the image space and Cartesian space are established (Fig. 6(a)). The plane $\mu$ consists of the centerline of the hole in the Cartesian space, its projection in the image plane and two projection rays. The plane $\pi$ consists of the centerline of the plug in the Cartesian space, its projection in the image plane and two projection rays. Hence, the direction of the projection line in the mobile image plane needs to be aligned to keep $\mu$ and $\pi$ parallel. During the process of alignment, the hole is motionless, while the peg is moving. Thus, we define,

$$\boldsymbol{s}_{\text{Mcoplanar}}(t) = [\rho_{Mp}(t), \theta_{Mp}(t)]^T, \boldsymbol{s}^*_{\text{Mcoplanar}} = [\rho_{Mh}, \theta_{Mh}]^T \tag{14}$$

where $\boldsymbol{s}_{\text{Mcoplanar}}(t)$ denotes the co-planar feature, which is the centerline feature of the peg. $\boldsymbol{s}^*_{\text{Mcoplanar}}$ denotes the desired co-planar feature, which is the centerline feature of the hole.

2. Co-linear constraint: The alignment of pitch angle

Under the co-planar constraint, the co-linear constraint is established in the fixed camera space, which corresponds to the pitch angle alignment (Fig. 6(b)). Similar to the co-planar constraint, we define,

$$\boldsymbol{s}_{\text{Fcolinear}}(t) = [\rho_{Fp}(t), \theta_{Fp}(t)]^T, \boldsymbol{s}^*_{\text{Fcolinear}} = [\rho_{Fh}, \theta_{Fh}]^T \tag{15}$$

where $\boldsymbol{s}_{\text{Fcolinear}}(t)$ denotes the co-linear feature, which is the centerline feature of the peg, and $\boldsymbol{s}^*_{\text{Fcolinear}}$ denotes the desired co-linear feature, which is the centerline feature of the hole.

3. Co-point constraint: The alignment of rotation angle and the terminal point

Under the above two constraints, the co-point constraint, both in the mobile and fixed image planes, will be considered in correspondence to the alignment of the rotation angle and the coordinate of the

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

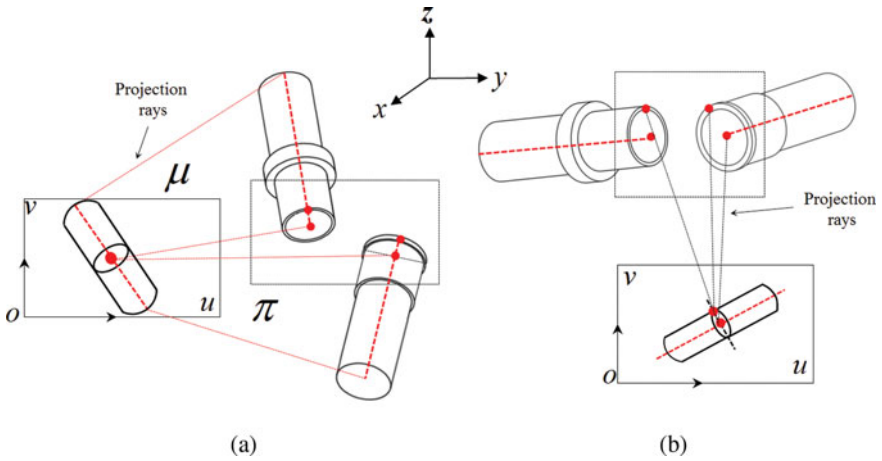



Fig. 6. Visual constraints between the peg and hole: (a) in the mobile image plane and (b) in the fixed image plane.

terminal point (Fig. 6). Similarly, we define,

$$\boldsymbol{s}_{\text{Mcopoint}}(t) = [k_{Mp}(t), n_{Mp}(t)]^T, \boldsymbol{s}^*_{\text{Mcopoint}} = [k_{Mh}, n_{Mh}]^T \tag{16}$$

$$\boldsymbol{s}_{\text{Fcopoint}}(t) = [k_{Fp}(t), n_{Fp}(t)]^T, \boldsymbol{s}^*_{\text{Fcopoint}} = [k_{Fh}, n_{Fh}]^T \tag{17}$$

where $\boldsymbol{s}_{\text{Mcopoint}}(t)$, $\boldsymbol{s}_{\text{Fcopoint}}(t)$ denote the co-point features, which are the terminal and the tip point features of the plug in the two image planes. $\boldsymbol{s}^*_{\text{Mcopoint}}$, $\boldsymbol{s}^*_{\text{Fcopoint}}$ denote the desired co-point features, which are the terminal and the tip point features of hole in the two image planes.

## 5. Control Strategy

In this section, the control strategies of the full task are presented. The desired trajectories of the two end-effectors' features in Section 3 and the visual constraints between the two objects in Section 4 will be used to realize the relative pose alignment of the two objects.

### *5.1. Desired trajectories tracking of end-effector features*

In step 1, it needs to regulate the two end-effectors' features to the desired features at the pre-alignment pose synchronously. The left end-effector is taken as the example, $j$ feature points ($j > 3$) are fixed on the end-effector, thus the feature can be denoted by $\boldsymbol{s}_L = (\boldsymbol{s}_1^{\mathrm{T}}\ \boldsymbol{s}_2^{\mathrm{T}} \cdots \boldsymbol{s}_j^{\mathrm{T}})^{\mathrm{T}}$, and the desired features $\boldsymbol{s}^*_L$ are given in Section 3. Hence, the error of the image feature point $i$ is defined in the fixed camera space $\boldsymbol{e}_{Li} = \boldsymbol{s}_{Li} - \boldsymbol{s}^*_{Li}$ .The time derivative of $\boldsymbol{s}_{Li}$ is given,

$$\dot{\boldsymbol{s}}_{Li} = \boldsymbol{L}_{Li} \cdot (\boldsymbol{v}^F - \boldsymbol{v}^F_{Li}) \tag{18}$$

with

$$\boldsymbol{L}_{Li}(u, v, z) = \begin{bmatrix} -f/z_i & 0 & u_i/z_i & u_i v_i/f & -(f + u_i{}^2/f) & v_i \\ 0 & -f/z_i & v_i/z_i & f + v_i{}^2/f & -u_i v_i/f & -u_i \end{bmatrix} \in R^{2\times 6} \tag{19}$$

where $f$ is the focal length, $(u, v)$ are the image coordinate. In addition, $\boldsymbol{v}^F$ and $\boldsymbol{v}^F_{Li}$ in (18) are the velocity screws with respect to $\Sigma_F$ . In this case, the fixed camera screw $\boldsymbol{v}^F$ is zero. The expression of $\boldsymbol{v}^F_{Li}$ is,

$$\boldsymbol{v}^F_{Li} = \boldsymbol{W}^F_B \cdot \boldsymbol{v}^B_{Li} \tag{20}$$

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

with $\boldsymbol{W}_B^F = \begin{bmatrix} \boldsymbol{R}_B^F & [\boldsymbol{t}_B^F]_\times \boldsymbol{R}_B^F \\ 0_{3\times3} & \boldsymbol{R}_B^F \end{bmatrix}$. Where $\boldsymbol{R}_B^F \in R^{3\times3}$ and $\boldsymbol{t}_B^F \in R^{3\times1}$ are respectively the rotation and translation between $\Sigma_B$ and $\Sigma_F$. $[\boldsymbol{t}_B^F]_\times$ denotes a skew-symmetric matrix. $\boldsymbol{v}_{Li}^B$ is the velocity screw of the end-effector with respect to $\Sigma_B$ , the expression is,

$$\boldsymbol{v}_{Li}^B = \boldsymbol{J}_L \cdot \dot{\boldsymbol{q}}_L \tag{21}$$

where $\boldsymbol{J}_L \in \boldsymbol{R}^{6\times7}$ , $\dot{\boldsymbol{q}}_L \in \boldsymbol{R}^7$. Then, we have

$$\dot{\boldsymbol{s}}_{Li} = \boldsymbol{J}_{Lsi}\dot{\boldsymbol{q}}_L, \text{ with } \boldsymbol{J}_{Lsi} = -\boldsymbol{L}_{Li}\boldsymbol{W}_B^F\boldsymbol{J}_L \tag{22}$$

$\boldsymbol{J}_{Lsi} \in \boldsymbol{R}^{2\times7}$ denotes the task Jacobian matrix of the feature point $i$ on the left end-effector. Considering all $j$ feature points, the task Jacobian matrix can be expressed by $\boldsymbol{J}_{Ls} = (\boldsymbol{J}_{Ls1}^{\mathrm{T}}\ \boldsymbol{J}_{Ls2}^{\mathrm{T}} \cdots \boldsymbol{J}_{Lsj}^{\mathrm{T}})^{\mathrm{T}}$, $\boldsymbol{J}_L \in \boldsymbol{R}^{2j\times7}$. Hence, the time derivative of all point features is given as

$$\dot{\boldsymbol{s}}_L = \boldsymbol{J}_{Ls}\dot{\boldsymbol{q}}_L, \text{ with } \boldsymbol{J}_{Ls} = -\boldsymbol{L}_L\boldsymbol{W}_B^F\boldsymbol{J}_L \tag{23}$$

where $\boldsymbol{L}_L = (\boldsymbol{L}_{L1}^T\boldsymbol{L}_{L2}^T \cdots \boldsymbol{L}_{Lj}^T)^T$, $(j > 3)$. Let $\boldsymbol{e}_L = \boldsymbol{s}_L - \boldsymbol{s}_L^*$ , and the control law is designed based on the task function in,

$$\dot{\boldsymbol{q}}_L = -\lambda_L \boldsymbol{J}_{Ls}^+ \boldsymbol{e}_L \tag{24}$$

where $\lambda_L$ is a positive gain, $e$ is the task function, $\boldsymbol{J}_{Ls}$ is the task Jacobian of the left arm, and $\boldsymbol{J}_{Ls}^+$ is the Moore–Penrose inverse, ${\boldsymbol{J}_{Ls}}^+ = (\boldsymbol{J}_{Ls}^T\boldsymbol{J}_{Ls})^{-1}\boldsymbol{J}_{Ls}^T$.

With regard to the 7-DOF redundant manipulator, the dimensions in the joint space are larger than ones in the task space. Thus, a null space $\boldsymbol{N}$ $(\boldsymbol{q})$ (non-null) exists, where the manipulator is capable of self-motion. The relation between $\boldsymbol{N}$ $(\boldsymbol{q})$ and $\boldsymbol{J}$ can be written as$\boldsymbol{J} \cdot \boldsymbol{N} = 0$, where $\boldsymbol{N}(\boldsymbol{N}^T\boldsymbol{N})^{-1}\boldsymbol{N}^T = (\boldsymbol{I} - \boldsymbol{J}^+\boldsymbol{J})$. The control law of redundancy formalism is,

$$\dot{\boldsymbol{q}}_L = -\lambda_L {\boldsymbol{J}_{Ls}}^+ \boldsymbol{e}_L + (\boldsymbol{I} - \boldsymbol{J}_{Ls}^+\boldsymbol{J}_{Ls})\alpha \tag{25}$$

In order to ensure all joint angles are limited within the allowable range, the homogeneous solution is optimized in Eq. (25). Gradient projection method (GPM) is a commonly used local optimization scheme. Let $k_L\nabla\boldsymbol{H}(\boldsymbol{q})$ be instead of $\alpha$ in Eq. (25). We have

$$\dot{\boldsymbol{q}}_L = -\lambda_L {\boldsymbol{J}_{Ls}}^+ \boldsymbol{e}_L + k_L(I - {\boldsymbol{J}_{Ls}}^+\boldsymbol{J}_{Ls})\nabla\boldsymbol{H}(\boldsymbol{q}_L) \tag{26}$$

where $k_l$ is constant, and the value of $k_l$ is limited by the joint velocity, $\boldsymbol{H}(\boldsymbol{q}_L)$ denotes the cost function of joint-limit avoidance, $\nabla\boldsymbol{H}(\boldsymbol{q}_L)$ is the gradient vector of $\boldsymbol{H}(\boldsymbol{q}_L)$, and

$$\nabla\boldsymbol{H}(\boldsymbol{q}_L) = \left( \frac{\partial\boldsymbol{H}(\boldsymbol{q}_L)}{\partial\boldsymbol{q}_{L1}} \ \frac{\partial\boldsymbol{H}(\boldsymbol{q}_L)}{\partial\boldsymbol{q}_{L2}} \cdots \frac{\partial\boldsymbol{H}(\boldsymbol{q}_L)}{\partial\boldsymbol{q}_{Ln}} \right)^T.$$

In this paper, $\boldsymbol{H}(\boldsymbol{q}_L)$ is given by

$$\boldsymbol{H}(\boldsymbol{q}_L) = \left|\left| \frac{\boldsymbol{q}_L - \bar{\boldsymbol{q}}_L}{\boldsymbol{q}_{L\max} - \boldsymbol{q}_{L\min}} \right|\right|_p \tag{27}$$

where $\boldsymbol{q}_{Li\min}, \boldsymbol{q}_{Li\max}$ are the lower and upper joint limits of $\boldsymbol{q}_{Li}$; $\bar{\boldsymbol{q}}_{Li}$ denotes the domain average of $q_{Li}$, and $p$ $(p \geq 2)$ is the norm, $p = 6$. Similarly, Let $\boldsymbol{e}_{R1} = \boldsymbol{s}_R - \boldsymbol{s}_R^*$ the control law of the right arm can be obtained as follows:

$$\dot{\boldsymbol{q}}_R = -\lambda_{R1} {\boldsymbol{J}_{Rs1}}^+ \boldsymbol{e}_{R1} + k_{R1}(\boldsymbol{I} - {\boldsymbol{J}_{Rs1}}^+\boldsymbol{J}_{Rs1})\nabla\boldsymbol{H}(\boldsymbol{q}_R) \tag{28}$$

where $\boldsymbol{e}_{R1}$ is the errors and $\boldsymbol{J}_{Rs1}$ is the Jacobian of right arm in step 1.

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

During the full task of dual-arm manipulation, the hole gripped by the left end-effector will be motionless when the pre-alignment pose of the left end-effector is achieved, therefore the objective task of the left arm will only have been completed by controlling the feature point of left end-effector. However, the plug gripped by the right end-effector will still keep moving from the pre-alignment pose to the accurate alignment pose between the two objects. In next part, the object feature is introduced additionally to design the control law of the right arm.

*5.2. Pose alignment of object features*

The movement of the right arm will be as follows: first, the pose of the right end-effector will be controlled in Eq. (29) as to attain the pre-alignment pose of the end-effector features in the fixed camera space; second, the position and orientation of the plug will be adjusted to the accurate alignment pose by observing the four visual features of the hole, both in the fixed and mobile camera spaces. Based on the previous discussions, the task function of the right arm is given,

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_{R2} \\ \boldsymbol{e}_{R3} \\ \boldsymbol{e}_{R4} \\ \boldsymbol{e}_{R5} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{\text{Mcoplanar}} - \boldsymbol{s}^*_{\text{Mcoplanar}} \\ \boldsymbol{s}_{\text{Fcolinear}} - \boldsymbol{s}^*_{\text{Fcolinear}} \\ \boldsymbol{s}_{\text{Mcopoint}} - \boldsymbol{s}^*_{\text{Mcopoint}} \\ \boldsymbol{s}_{\text{Fcopoint}} - \boldsymbol{s}^*_{\text{Fcopoint}} \end{bmatrix} \tag{29}$$

Each subtask $\boldsymbol{e}_i (i = 2, \cdots, 5)$ is to be performed in sequence, thus the active error is introduced[30]

$$\boldsymbol{e}_A = \boldsymbol{A} \cdot \boldsymbol{e} \tag{30}$$

where the component $a$ of $\boldsymbol{A}$ determines whether the image value is active or inactive. If $a = 1$ , the feature is active, and if $a = 0$ the feature is inactive. The active matrix $\boldsymbol{A}$ is,

$$\boldsymbol{A} = \text{diag}\{\boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{A}_4, \boldsymbol{A}_5\} \tag{31}$$

with

$$\begin{aligned} &\boldsymbol{A}_2 = 0_{2\times 2} \rightarrow \boldsymbol{I}_{2\times 2}; \\ &\boldsymbol{A}_3 = 0_{2\times 2} \rightarrow \boldsymbol{I}_{2\times 2}, \quad when\ \boldsymbol{e}_{R2} = 0; \\ &\boldsymbol{A}_4 = 0_{3\times 3} \rightarrow \boldsymbol{I}_{3\times 3}, \quad when\ \boldsymbol{e}_{R3} = 0; \\ &\boldsymbol{A}_5 = 0_{3\times 3} \rightarrow \boldsymbol{I}_{3\times 3}, \quad when\ \boldsymbol{e}_{R4} = 0. \end{aligned}$$

According to the above conditions, all subtasks can be carried out in sequence. The derivative of Eq. (29) is,

$$\dot{\boldsymbol{e}}_A = A\dot{\boldsymbol{e}} = A\left(\dot{\boldsymbol{s}}^T_{\text{Mcoplanar}} \dot{\boldsymbol{s}}^T_{\text{Fcolinear}} \dot{\boldsymbol{s}}^T_{\text{Mcopoint}} \dot{\boldsymbol{s}}^T_{\text{Fcopoint}}\right)^T \tag{32}$$

In addition, the time derivative of the object feature in the mobile image plane is given by

$$\dot{\boldsymbol{s}}_M = \boldsymbol{L}_M(\boldsymbol{v}^M - \boldsymbol{v}^M_R) \tag{33}$$

where $\boldsymbol{v}^M = \boldsymbol{J}_M \dot{\boldsymbol{q}}_R$, $\boldsymbol{v}^M_R = \boldsymbol{W}^M_R \boldsymbol{J}_R \dot{\boldsymbol{q}}_R$ , and $\boldsymbol{J}_M$ is the Jacobian of the mobile camera, which can be obtained by the coordinate relationship $\boldsymbol{v}^B_M = \boldsymbol{v}^B_R \boldsymbol{v}^R_M$ . Thus, we have

$$\dot{\boldsymbol{s}}_M = \boldsymbol{L}_M(\boldsymbol{J}_M - \boldsymbol{W}^M_R \boldsymbol{J}_R)\dot{\boldsymbol{q}}_R \tag{34}$$

And the time derivatives of $\boldsymbol{s}_{\text{Mcoplanar}}$, $\boldsymbol{s}_{\text{Mcopoint}}$ in the mobile image plane are given as follows:

$$\begin{cases} \dot{\boldsymbol{s}}_{\text{Mcoplanar}} = \boldsymbol{L}_{\text{Mcoplanar}}(\boldsymbol{J}_M - \boldsymbol{W}^M_R \boldsymbol{J}_R)\dot{\boldsymbol{q}}_R \\ \dot{\boldsymbol{s}}_{\text{Mcopoint}} = \boldsymbol{L}_{\text{Mcopoint}}(\boldsymbol{J}_M - \boldsymbol{W}^M_R \boldsymbol{J}_R)\dot{\boldsymbol{q}}_R \end{cases} \tag{35}$$

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

Similarly, the time derivative of the feature in the fixed image plane is given as follows:

$$\dot{\boldsymbol{s}}_F = -\boldsymbol{L}_F \boldsymbol{W}_R^F \boldsymbol{J}_R \dot{\boldsymbol{q}}_R \tag{36}$$

Then, the time derivatives of $\boldsymbol{s}_{\text{Fcolinear}}$, $\boldsymbol{s}_{\text{Fcopoint}}$ in the fixed image space are given as follows:

$$\begin{cases} \dot{\boldsymbol{s}}_{\text{Fcolinear}} = -\boldsymbol{L}_{\text{Fcolinear}} \boldsymbol{W}_R^F \boldsymbol{J}_R \dot{\boldsymbol{q}}_R \\ \dot{\boldsymbol{s}}_{\text{Fcopoint}} = -\boldsymbol{L}_{\text{Fcopoint}} \boldsymbol{W}_R^F \boldsymbol{J}_R \dot{\boldsymbol{q}}_R \end{cases} \tag{37}$$

where $\boldsymbol{L}_{\text{Mcoplanar}}$, $\boldsymbol{L}_{\text{Fcolinear}}$, $\boldsymbol{L}_{\text{Mcopoint}}$, and $\boldsymbol{L}_{\text{Fcopoint}}$ are the interaction matrices of the four alignment features. Next, the concrete expression of those interaction matrices will be given.

The interaction matrices of the point features are denoted by $\boldsymbol{L}_n(u_n, v_n, z_n)$ , $\boldsymbol{L}_c(u_c, v_c, z_c)$, and $\boldsymbol{L}_t(u_t, v_t, z_t)$ , the expressions of which have been given in Eq. (19). The interaction matrix of the line feature is given as follows:

$$\boldsymbol{L}_{\rho\theta} = \begin{bmatrix} -1 & v_c \cdot \cos(\theta) - u_c \cdot \sin(\theta) \\ -1 & v_t \cdot \cos(\theta) - u_t \cdot \sin(\theta) \end{bmatrix}^{-1} \begin{bmatrix} -\cos(\theta) & -\sin(\theta) & 0 & 0 \\ 0 & 0 & -\cos(\theta) & -\sin(\theta) \end{bmatrix} \begin{bmatrix} \boldsymbol{L}_c \\ \boldsymbol{L}_t \end{bmatrix} \tag{38}$$

And the feature $k$ of the terminal point position is

$$\boldsymbol{L}_k = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & 0 & (-u_t \cdot \cos(\theta) - v_t \cdot \sin(\theta)) \end{bmatrix} \begin{bmatrix} \boldsymbol{L}_t \\ \boldsymbol{L}_{\rho\theta} \end{bmatrix} \tag{39}$$

Therefore, the interaction matrices of the co-planar feature $\boldsymbol{L}_{\text{Mcoplanar}} = \boldsymbol{L}_{M\rho\theta}$ , the co-linear feature $\boldsymbol{L}_{\text{Fcolinear}} = \boldsymbol{L}_{F\rho\theta}$, and the co-point features $\boldsymbol{L}_{\text{Mcopoint}} = [\, \boldsymbol{L}_{Mk}^T \; \boldsymbol{L}_{Mn}^T \,]^T$, $\boldsymbol{L}_{\text{Fcopoint}} = [\, \boldsymbol{L}_{Fk}^T \; \boldsymbol{L}_{Fn}^T \,]^T$ can be obtained. Combining Eqs. (35) and (37), the relationship between the right joint space and the two camera spaces are obtained as follows:

$$\left( \dot{\boldsymbol{s}}_{\text{Mcoplanar}}^T \dot{\boldsymbol{s}}_{\text{Fcolinear}}^T \dot{\boldsymbol{s}}_{\text{Mcopoint}}^T \dot{\boldsymbol{s}}_{\text{Fcopoint}}^T \right)^T = \boldsymbol{A} \boldsymbol{J}_{Rs2} \dot{\boldsymbol{q}}_R \tag{40}$$

with

$$\boldsymbol{J}_{Rs2} = \begin{bmatrix} -\boldsymbol{L}_{\text{Mcoplanar}} \boldsymbol{W}_R^M & \boldsymbol{L}_{\text{Mcoplanar}} \\ -\boldsymbol{L}_{\text{Fcolinear}} \boldsymbol{W}_R^F & 0 \\ -\boldsymbol{L}_{\text{Mcopoint}} \boldsymbol{W}_R^M & \boldsymbol{L}_{\text{Mcopoint}} \\ -\boldsymbol{L}_{\text{Fcopoint}} \boldsymbol{W}_R^F & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{J}_R \\ \boldsymbol{J}_M \end{bmatrix}, \quad \boldsymbol{J}_{Rs2} \in R^{10\times 7}.$$

where $\boldsymbol{J}_{Rs2}$ is defined as the multi-tasks image Jacobian of right arm. Let $\boldsymbol{e}_H$ be instead of $\boldsymbol{e}$ , then the control law of the right arm for the object alignment is given,

$$\dot{\boldsymbol{q}}_R = -\lambda_{R2} (\boldsymbol{A} \boldsymbol{J}_{Rs2})^+ \boldsymbol{A} \cdot \boldsymbol{e} + k_{R2} \left( \boldsymbol{I} - \boldsymbol{J}_{Rs2}^+ \boldsymbol{J}_{Rs2} \right) \nabla \boldsymbol{H}(\boldsymbol{q}_R) \tag{41}$$

So far, the dual-arm coordinated control laws in Eqs. (26), (28), and (41) have been obtained to attain the accurate alignment of the peg and hole with the large initial pose errors.

## 6. Experiments and Results

### *6.1. Experiment setup*

In order to validate the proposed strategy, two experiments that the alignment of plug and socket and cup and lid are carried out using a MOTOMAN dual-arm robot. The experimental setup is shown in Fig. 7. Both cameras are calibrated offline by the Tsai algorithm. The mobile camera pose, with respect to the right end-effector frame $\boldsymbol{W}_R^M$, and the fixed camera pose, with respect to the robot base frame $\boldsymbol{W}_B^F$ , are roughly calibrated. Moreover, the coordinates of the two end-effectors' marks with respect to the end-effector frame are measured, respectively. In step 1, the fixed camera observes the

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

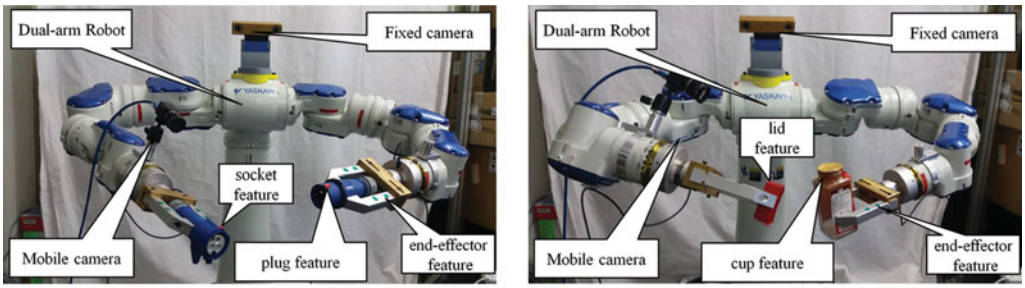



Fig. 7. Experimental setup: (a) plug-and-socket alignment and (b) cup-and-lid alignment.

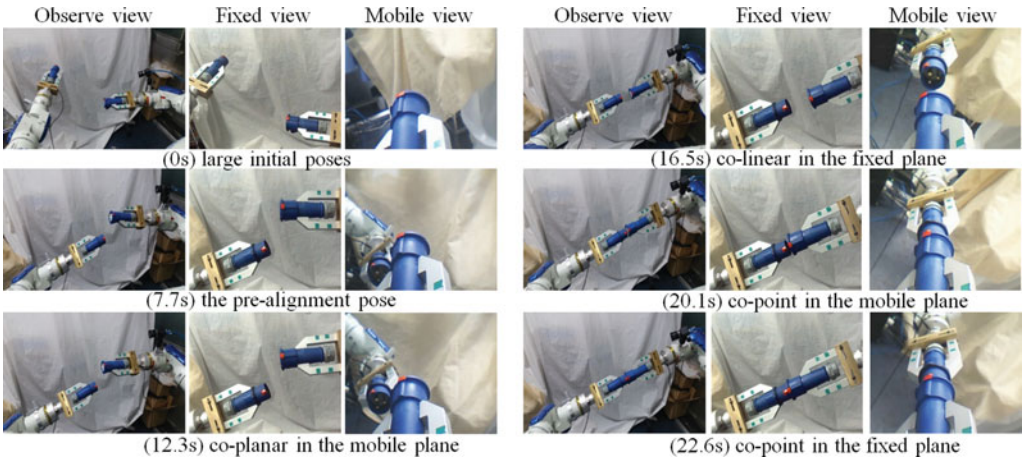



Fig. 8. Experiment snapshots for the relative pose alignment from three viewpoints that include the observing camera, the fixed camera and the mobile camera. At 0s: the plug and socket are placed at the initial positions; 7.7s: the pre-alignment poses in the fixed camera space; 12.3s: co-planar in the mobile camera space; 16.5s: co-linear in the fixed camera space; 20.1s: co-point in the mobile camera space; 22.6s: co-point in the fixed camera space.

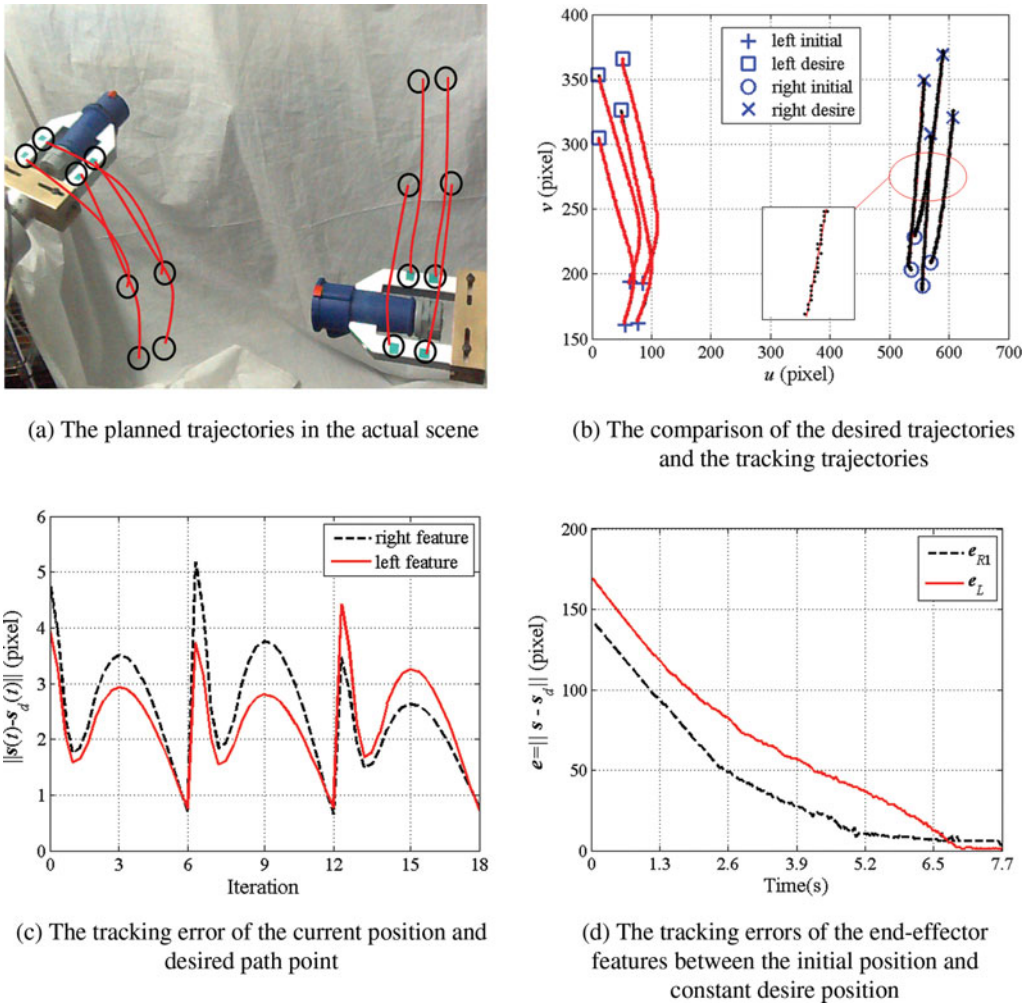
features on the two end-effectors which are composed of four green marks. All the centroids of the marks are extracted by the image processing algorithm based on the color space.

### *6.2. Plug-and-socket alignment*

Both the plug and socket are gripped by the end-effectors in parallel where the center lines of the object and end-effector are kept co-linear. The distance between the right arm end-point and the terminal point of the plug is 245 mm, and the distance between the left arm end-point and the terminal point of the socket is 252 mm. The position and orientation of the plug and socket in the Cartesian space can be got by the object model and robot kinematics model. The large initial pose errors are given as follows: position $\Delta x = 81.2$ mm, $\Delta y = 280.6$ mm, $\Delta z = -74.3$ mm, average error 145.4 mm, orientation $\Delta Rx = 44.3^\circ$, $\Delta Ry = -86.2^\circ$, $\Delta Rz = -0.9^\circ$, and average error $43.8^\circ$.

The experiment snapshots from three viewpoints are shown in Fig. 8. An extra camera is placed in the workspace to observe the experiment process, and the scenes in the fixed and mobile cameras are also shown. The full task is divided into five subtasks. From 0s to 7.7s (the first step of the full task), the two end-effectors are controlled to achieve the pre-alignment pose through 2D visual servoing, which is performed by observing the marks fixed on the end-effectors. Particularly at 0s, due to the large initial relative pose errors of the plug and socket, some features cannot be observed clearly in the fixed view (the feature $n$ and $\rho$), while the socket cannot even be observed in the mobile view. This is the reason why the end-effector features need to be observed instead of the object features under the large initial relative pose error of the plug and socket. In the second step of the task, from 7.7s to 12.3s, the co-planar features are considered to make the plug-and-socket co-planar in the mobile image plane; from 12.3s to 16.5s, the co-linear features are considered to make the plug-and-socket

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press




(a) The planned trajectories in the actual scene

(b) The comparison of the desired trajectories and the tracking trajectories

(c) The tracking error of the current position and desired path point

(d) The tracking errors of the end-effector features between the initial position and constant desire position

Fig. 9. The behaviors of the end-effector features in the image space: (a) The planned trajectories in the actual scene. (b) The comparison of the desired trajectories and the tracking trajectories. (c) The tracking error of the current position and desired path point. (d) The tracking errors of the end-effector features between the initial position and constant desire position.

co-linear in the fixed image plane; from 16.5s to 20.1s, the co-point features are considered to make the plug-and-socket co-point in the mobile image plane; from 20.1s to 22.6s, the co-point features are considered to make the plug-and-socket co-point in the fixed image plane.

In step 1 (0s–7.7s), the viusal path planning method in Section 3, is applied to generate the desired reference trajectories of the two end-effector features (8 feature points). In this experiment, 50 intermediate image coordinates of each feature point ($\sigma = 50$) are used to connect their initial image coordinates and constant desired image coordinates. The desired reference trajectories of 8 feature points, generated by the proposed method in the fixed image space, overlaided on the actual scene are shown as Fig. 9(a). Accordingly, as shown in Fig. 9(b), the followed trajectories are generated by moving two arms simultaneously using 2D visual servoing in Section 5.1. A comparison in Fig. 9(b) indicates that the error of the planned and followed trajectories at any intant will be smaller than 2 pixels. On the other hand, it should be ensured that the current image coordinates of each feature point can be close to their respective 50 intermediate reference image coordinates in sequence. The intermediate reference image coordinates which are nearest to the current image coordinates are taken as the current reference coordinates, and when the errors between them are regulated below the specified threshold, the next intermediate reference image's coordinates would become the current reference ones. Note that the errors between the current and reference image coordinates should be small enough to guarantee the feasibility of intermediate reference image coordinates. The experiment

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

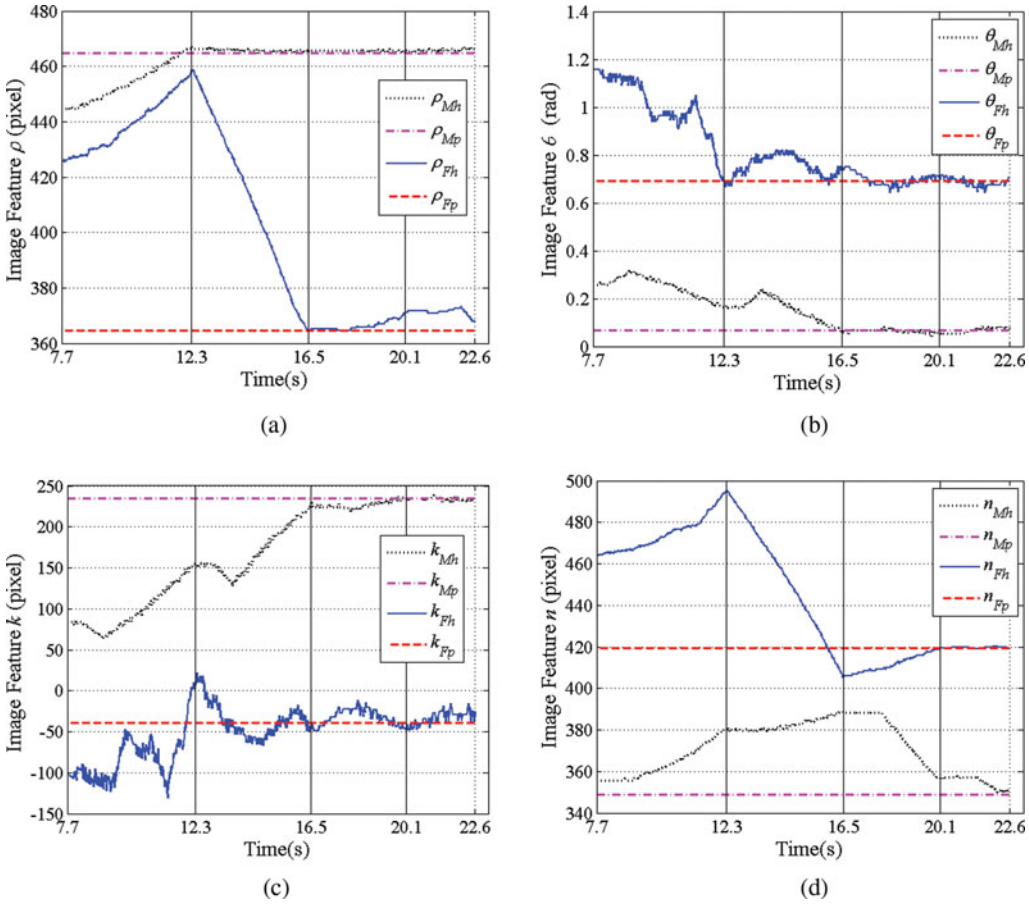



Fig. 10. The behaviors of the plug-and-socket features in the image space: (a) shows the image features $\rho$, (b) shows the image features $\theta$, (c) shows the image features $k$, and (d) shows the image features $n$.

results showing the errors in the first three regulation periods of the two end-effector features are shown in Fig. 9(c). As we can see that six iterations are needed to regulate the current image features to the current reference ones. The norms of these errors at the end of each regulation period are under 1 pixel. Furthermore, the task function errors $e_{R1}, e_{L1}$ in Eqs. (26), (28) between the current and constant desired image features gradually converge to zero in Fig. 9(d). Finally, the initial image features of the two end-effector features can be regulated to their corresponding desired features simultaneously.

In step 2 (7.7s–22.6s), the time change curves of the eight features in the two image planes are shown in Fig. 10. From 7.7s to 12.3s, the subtask is to make the co-planar feature $\boldsymbol{s}_{\text{Mcoplanar}} = [\rho_{Mp}, \theta_{Mp}]^T$ converge to the desired feature $\boldsymbol{s}^*_{\text{Mcoplanar}} = [\rho_{Mh}, \theta_{Mh}]^T$. As shown Fig. 10(a) and (b), at 12.3s, $\rho_{Mp}$ has achieved $\rho_{Mh}$ within a small error (< 2 pixels). Similarly, $\theta_{Mp}$ has achieved the $\theta_{Mh}$ within the error of 6.8°. In the later process, the co-planar constraint was still considered, where $\rho_{Mp}$ well converge to $\rho_{Mh}$. However, $\theta_{Mp}$ had large fluctuations, with the largest error reaching 4.5° in the co-linear process, and converged to $\theta_{Mh}$ again in the co-point process. These fluctuations were generated because of measurement errors of the image coordinates, including the object centroid and the tip points in the mobile image plane. These errors caused a large computation error in $\theta_M$, in other words, $\theta_M$ was sensitive to the measurement error of the image coordinates. From 12.3s to 16.5s, the collinear feature $\boldsymbol{s}_{\text{Fcolinear}} = [\rho_{Fp}, \theta_{Fp}]^T$ was required to achieve the desired feature $\boldsymbol{s}^*_{\text{Fcolinear}} = [\rho_{Fh}, \theta_{Fh}]^T$ (Fig. 10(a) and (b)). At 16.5s, the feature $\boldsymbol{s}_{\text{Fcolinear}}$ converged to $\boldsymbol{s}^*_{\text{Fcolinear}}$ with the small error of 1 pixel, 5.73°. In the later subtasks, the behavior was such that $\theta_{Fp}$ had only slight fluctuations around $\theta_{Fh}$, accordingly, the error between $\rho_{Fp}$ and $\rho_{Fh}$ had a slight rise during the co-point process, but the final error is limited within 2 pixels. From 16.5s to 20.1s, the co-point linear feature in the mobile camera $\boldsymbol{s}_{\text{Mcopoint}} = [k_{Mp}, n_{Mp}]^T$ is observed to achieve the desired feature $\boldsymbol{s}^*_{\text{Mcopoint}} = [k_{Mh}, n_{Mh}]^T$

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

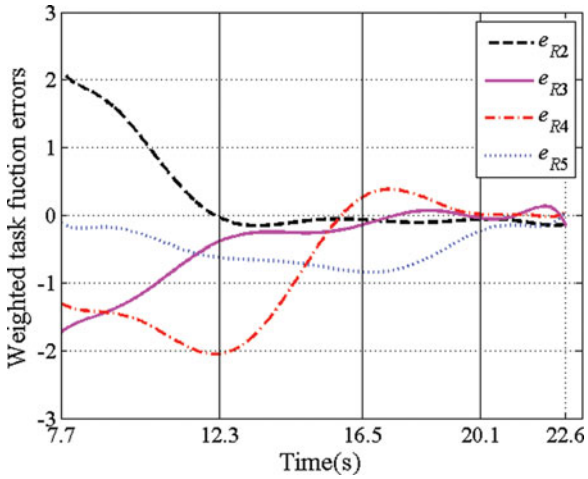



Fig. 11. The evolution of the task function which includes four subtasks errors.

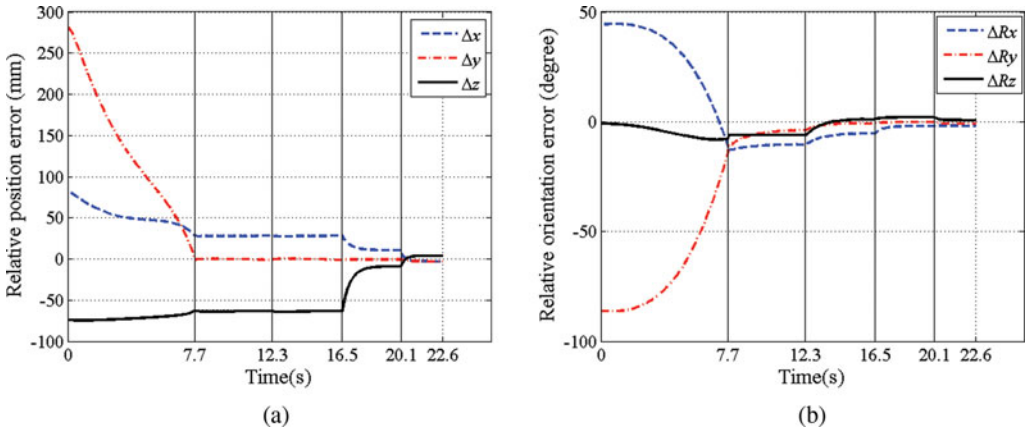



Fig. 12. The behaviors of the full task of the robot in the Cartesian space: (a) shows the relative position error of the tip point of the plug and socket and (b) shows the relative orientation error of the plug and socket.

(Fig. 10(c) and (d)). The image feature $n_{Mp}$ converged well to $n_{Mh}$, moreover, in the next subtask, only a slight change existed. However, $k_{Mp}$ had big fluctuations with convergence to $k_{Mh}$. According to Eq. (12), the value of $k_{Mp}$ is influenced by that of $\theta_{Mp}$, therefore the big fluctuations of $k_{Mp}$ came from $\theta_{Mp,}$ this was discussed in Fig. 10(b). From 20.1s to 22.6s, the co-point linear feature error of $\boldsymbol{s}_{\text{Fcopoint}} = [k_{Fp}, n_{Fp}]^T$ and $\boldsymbol{s}^*_{\text{Fcopoint}} = [k_{Fh}, n_{Fh}]^T$ in the fixed camera was controlled to converge to zero, as shown in Fig. 10(c) and (d). Finally, the change curves of $\boldsymbol{s}_{\text{Fcopoint}} = [k_{Fp}, n_{Fp}]^T$ converge to the ones of $\boldsymbol{s}^*_{\text{Fcopoint}} = [k_{Fh}, n_{Fh}]^T$ with a small error (<1 pixel).

The task function errors $\boldsymbol{e} = (\boldsymbol{e}_{R2}, \boldsymbol{e}_{R3}, \boldsymbol{e}_{R4}, \boldsymbol{e}_{R5})^T$ are presented in Fig. 11. From 7.7s to 12.3s, the first subtask $\boldsymbol{e}_{R2}$ (co-planar) is active and the norm of $\boldsymbol{e}_{R2}$ drops under a threshold (1 pixel) at 12.3s, the second subtask $\boldsymbol{e}_{R3}$ (co-linear) becomes active. When the norm of $\boldsymbol{e}_{R3}$ reached the threshold, 1 pixel, at 16.5s, the subtask $\boldsymbol{e}_{R4}$ (co-point in the mobile camera space) would start. At 16.5s, the value of $\boldsymbol{e}_{R4}$ becomes smaller than the chosen threshold and the last subtask $\boldsymbol{e}_{R5}$ (co-point in the fixed camera space) is launched. As we can see, finally all converge to zero, which indicates that the full task is achieved.

The behaviors of the robot in the Cartesian space and in the joint space are shown in Fig. 12. In the Cartesian space, the relative position and orientation change curves between the two tip points of the plug and socket is shown as Fig. 12(a) and (b). In the first step, the initial relative position error is $\Delta x = 81.2$ mm, $\Delta y = 280.6$ mm, $\Delta z = -74.3$ mm, and the initial relative orientation error is $\Delta Rx = 44.3°$, $\Delta Ry = -86.2°$, $\Delta Rz = -0.9°$. When the two end-effectors achieved the specific pre-alignment pose, which is set in advance through teaching by showing, the errors of all axes are

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

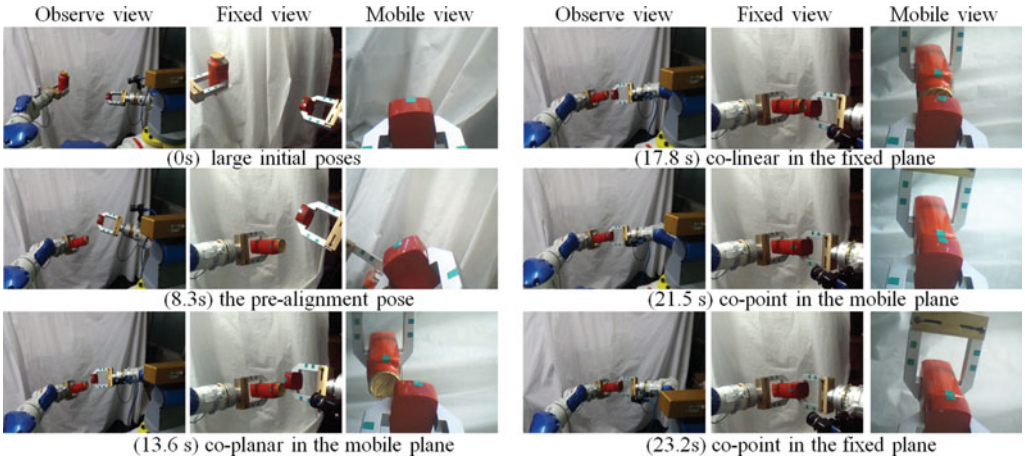



Fig. 13. Experiment snapshots for the relative pose alignment from three viewpoints that include the observing camera, the fixed camera and the mobile camera. At 0s: the cup and lid are placed at the large initial positions; 8.3.7s: the pre-alignment poses in the fixed camera space; 13.6s: co-planar in the mobile camera space; 17.8s: co-linear in the fixed camera space; 21.5s: co-point in the mobile camera space; 23.2s: co-point in the fixed camera space.

$\Delta x$ =28.2 mm, $\Delta y = -0.7$ mm, $\Delta z = -63.1$ mm, $\Delta Rx = -11.7^\circ$, $\Delta Ry = -13.6^\circ$, $\Delta Rz = -7.6^\circ$. In the second step, from 7.7s to 16.5s, by regulating the relative orientation error, the co-planar and co-linear constraints for the plug and socket are realized; from 16.5s to 22.6s, by regulating the relative position and orientation, the co-planar, co-linear, and co-point constraints are realized simultaneously. Finally, the plug and socket are successfully aligned, with the pose errors: position $\Delta x = -2.6$ mm, $\Delta y = -2.8$ mm, $\Delta z = 3.8$ mm, average error 3.1 mm, and orientation $\Delta Rx = -1.7^\circ$, $\Delta Ry = -0.9^\circ$, $\Delta Rz$ =0.7°, average error 1.1°, which have achieved the alignment accuracy that the position error of each axis < 5 mm, and the orientation error of each direction < 2°.

### *6.3. Cup-and-lid alignment*

The cup is gripped by the left end-effector in vertical, and the lid is gripped by the right end-effector in parallel, where the center lines of the lid and end-effector are kept co-linear. In step 2, the cup (peg) keeps motionless. The distance between the left arm end-point and the terminal point of the cup is 165 mm in parallel, 124 mm in vertical, and the distance between the right arm end-point and the terminal point of the lid is 155 mm. Then, the position and orientation of the cup and lid in the Cartesian space can be computed by the object model and robot kinematics model. The large initial pose errors are given as follows: position $\Delta x = 97.5$ mm, $\Delta y = 215.8$ mm, $\Delta z = 81.8$ mm, average 131.7 mm, and orientation $\Delta Rx = 43.6^\circ$, $\Delta Ry = -17.2^\circ$, $\Delta Rz = -0.4^\circ$, average 20.4°.

The experiment snapshots from three viewpoints are shown in Fig. 13. From 0s to 8.3s (step 1), only the end-effector features was observed, and it is unrelated to the objects. In step 2, from 8.3s to 13.6s, as shown in Fig. 14(a) and (b), the co-planar features $\boldsymbol{s}_{\text{Mcoplanar}} = [\rho_{Mh}(t), \theta_{Mh}(t)]^T$ (red line) was regulated to the desired feature $\boldsymbol{s}^*_{\text{Mcoplanar}} = [\rho_{Mp}, \theta_{Mp}]^T$ at the 13.6s. Next, the co-linear features $\boldsymbol{s}_{\text{Flinear}} = [\rho_{Fh}(t), \theta_{Fh}(t)]^T$ (blue line) converged to $\boldsymbol{s}^*_{\text{Flinear}} = [\rho_{Fp}, \theta_{Fp}]^T$ at 17.8s. Fig. 14(c) and (d) shows the co-point feature. At 21.5s, the tip points and terminal points were aligned in the mobile plane by regulating $\boldsymbol{s}_{\text{Mcopoint}}(t) = [k_{Mh}(t), n_{Mh}(t)]^T$ (blue line) to $\boldsymbol{s}^*_{\text{Mcopoint}}(t) = [k_{Mp}, n_{Mp}]^T$ and at 23.2s, regulating $\boldsymbol{s}_{\text{Fcopoint}}(t) = [k_{Fh}(t), n_{Fh}(t)]^T$ (blue line) to $\boldsymbol{s}^*_{\text{Fcopoint}}(t) = [k_{Fp}, n_{Fp}]^T$. Finally, the alignment task of cup and lid was achieved.

The errors of the position and orientation of cup and lid in the Cartesian space are shown in Fig. 15(a) and (b). The large initial relative position error is $\Delta x = 97.5$ mm, $\Delta y = 215.8$ mm, $\Delta z$ = 81.8 mm, average 131.7 mm, and the initial relative orientation error is $\Delta Rx = 43.6^\circ$, $\Delta Ry = -17.2^\circ$, $\Delta Rz = -0.4^\circ$, average 20.4°. Finally, the alignment accuracy is as follows: position $\Delta x = -3.1$ mm, $\Delta y = -0.8$ mm, $\Delta z = -4.2$ mm, average −2.7 mm, and orientation $\Delta Rx = -1.5^\circ$, $\Delta Ry = -0.4^\circ$, $\Delta Rz = -0.5^\circ$, average -0.8°, which have achieved the alignment accuracy that the position error of each axis < 5 mm, and the orientation error of each direction < 2°.

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

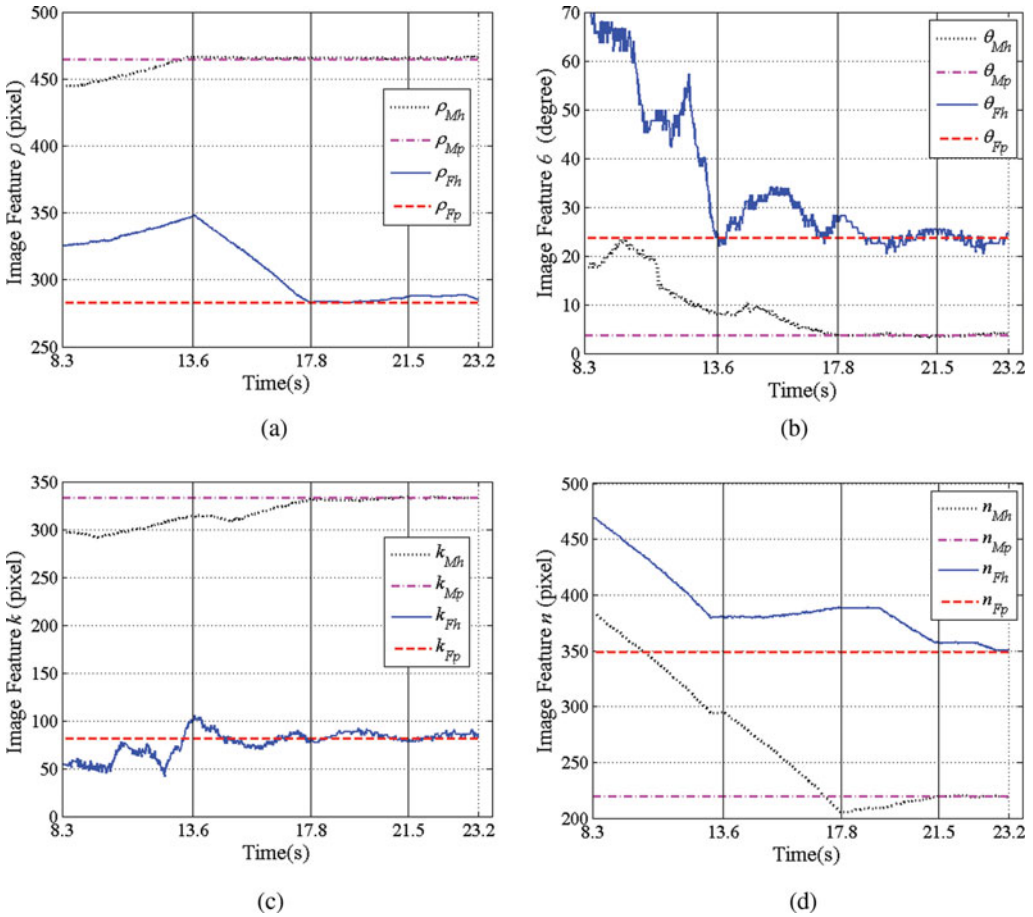



Fig. 14. The behaviors of the cup-and-lid features in the image space: (a) shows the image features $\rho$, (b) shows the image features $\theta$, (c) shows the image features $k$, and (d) shows the image features $n$.

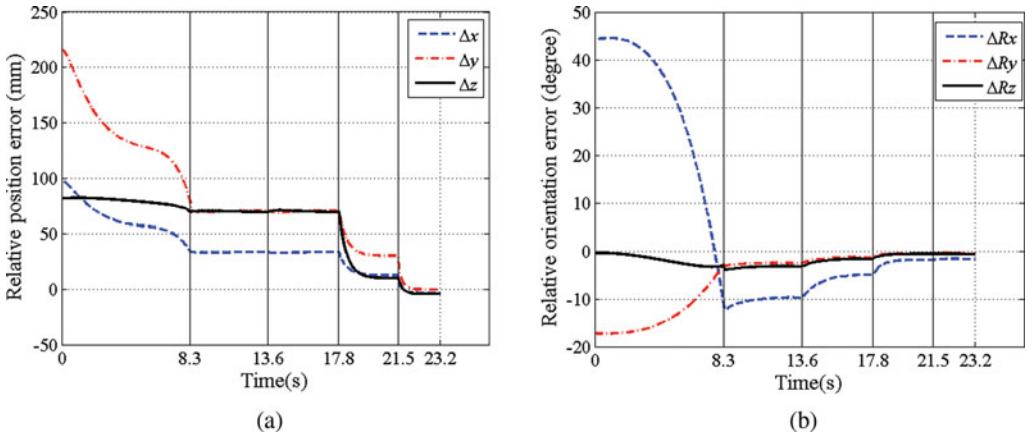



Fig. 15. The behaviors of the full task of the robot in the Cartesian space: (a) shows the relative position error of the terminal point of the cup and lid and (b) shows the relative orientation error of the cup and lid.

## 7. Conclusion

In order to solve the large initial pose uncertainty of a dual-arm manipulation. This paper developed a new multi-cameras visual servoing method that combines the visual path planning with multi-tasks IBVS method. The proposed method extends the conventional IBVS method to a large-scale application, and achieves the position and orientation control of the robot. The proposed method greatly enhanced the flexibility in a real dual-arm robot application.

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press

The two alignment tasks are performed using a MOTOMAN dual-arm robot platform. Experimental results indicate that the proposed strategy is effective to the pose alignment tasks. The experiment of the plug and socket with the large initial pose errors: the average errors of position 145.4 mm and orientation 43.8°, and successfully aligned with allowed pose errors: the average errors of position 3.1 mm and orientation 1.1°. The experiment of the cup and lid with large initial pose error: position average 131.7 mm, orientation average 20.4°, finally, the accuracy is average −2.7 mm, −0.8° within an allowable range.

**Acknowledgements**
This work is supported by the National Natural Science Foundation of China (Grant No. 61673134), and the Self-Planned Task (No. SKLRS201501C) of State Key Laboratory of Robotics and System (HIT).

**References**

1. J. Stuckler and S. S. Behnke, "Following Human Guidance to Cooperatively Carry a Large Object," *Proceedings of the IEEE International Conference on Humanoid Robots*, Piscataway, NJ: IEEE, Bled, Slovenia (Oct. 2011) pp. 218–223.
2. S. Y. Shin and C. H. Kim, "Human-like motion generation and control for humanoid's dual arm object manipulation," *IEEE Trans. Ind. Electron.* **62**(4), 2265–2276 (Apr. 2015).
3. Y. Ren, Y. C. Liu, M. H. Jin and H. Liu, "Biomimetic object impedance control for dual-arm cooperative 7-DOF manipulators," *Robot. Auton. Syst.* **75**, 273–287 (Jan. 2016).
4. P. Dauchez, P. Fraisse and F. Pierrot, "A Vision/Position/Force Control Approach for Performing Assembly Tasks with a Humanoid Robot," *Proceedings of the IEEE/RSJ International Conference on Humanoid Robots*, Piscataway, NJ: IEEE, Tsukuba, Japan (Dec. 2005) pp. 277–282.
5. K. Yamazaki, R. Oya, K. Nagahama, K. Okada and M. Inaba, "Bottom dressing by a dual-arm robot using a clothing state estimation based on dynamic shape changes," *Int. J. Adv. Robot. Syst.* **13**(5), 1–13 (Jan. 2016).
6. M. Beetz, U. Klank, A. Maldonado D. Pangercic and T. Ruhr, "Robotic Roommates Making Pancakes," *Proceedings of the IEEE International Conference on Humanoid Robots*, Piscataway, NJ: IEEE, Bled, Slovenia (Oct. 2011) pp. 529–536.
7. C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas and D. Kragic, "Dual arm manipulation: A survey," *Robot. Auton. Syst.* **60**, 1340–1353 (Oct. 2012).
8. D. Kragic and H. I. Christensen, "Survey on visual servoing for manipulation," Royal Inst. Technol., Stockholm, Sweden, Tech. Rep. ISRN KTH/NA/P-02/01-SE, (2002).
9. S. Hutchinson, G. D. Hager and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.* **12**(5), 651–670 (Feb. 1996).
10. O. Kermorgant and F. Chaumette, "Multi-sensor Data Fusion in Sensor-based Control: Application to Multi-camera Visual Servoing," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Piscataway, NJ: IEEE, ShangHai, China (May 2011) pp. 4518–4523.
11. E. Malis, F. Chaumette and S. Boudet, "Multi-cameras Visual Servoing," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Piscataway, NJ: IEEE, San Francisco, CA (Apr. 2000) pp. 3183–3188.
12. F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robot, Automat. Mag.* **13**(4), 82–90 (Dec. 2006).
13. F. Chaumette and S. Hutchinson, "Visual servo control, part ii: Advanced approaches," *IEEE Robot, Automat. Mag.* **14**(1), 109–118 (Mar. 2007).
14. V. Lippiello, B. Siciliano and L. Villani, "Position-based Visual Servoing in Industrial Multi-robot Cells Using a Hybrid Camera Configuration," *IEEE Trans. Robot.* **23**(1), 73–86 (Feb. 2007).
15. R. L. A. Shauri and K. Nonami, "Assembly manipulation of small objects by dual-arm manipulator," *Assem. Autom.* **31**(3), 263–274 (2011).
16. N. Vahrenkamp, C. Boge, K. Welke, T. Asfour, J. Walter and R. Dillmann, "Visual Servoing for Dual Arm Motions on a Humanoid Robot," *Proceedings of the IEEE/RSJ International Conference on Humanoid Robots*, Paris, France (Dec. 2009) pp. 208–214.
17. K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic and R. Dillmann, "Grasping Known Objects with Humanoid Robots; a Box-based Approach," *Proceedings of the International Conference on Advanced Robitics (ICAR)*, Munich, Germany (2009) pp. 179–184.
18. Y. Shen, D. Xu, M. Tan and J. Z. Yu, "Mixed visual control methods for robots with self-calibrated stereo rig," *IEEE Trans. Instrum. Meas.* **59**(2), 470–479 (Feb. 2010).
19. S. Han, W. See, J. Lee, M. Lee and H. Hashimoto, "Image-based Visual Servoing Control of a Scara Type Dual-arm Robot," *Proceedings of the IEEE International Symposium Industrial Electronics (ISIE)*, Cholula, Mexico (2000) vol. 2, pp. 517–522.

20. P. Hynes, G. Dodds and A. J. Wilkinson, "Uncalibrated Visual Servoing of a Dual Arm Robot for Mis Suturing," *Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Pisa, Italy (Feb. 2006) pp. 420–425.
21. D. Kruse, J. T. Wen and R. J. Radke, "A sensor-based dual-arm tele-robotic system," *IEEE Trans. Autom. Sci. Eng.* **12**(1), 4–18 (Jan. 2015).
22. E. Zereik, A. Sorbara, G. Casalino and F. Didot, "Autonomous Dual-arm Mobile Manipulator Crew Assistant for Surface Operations: Force/vision –Guided Grasping," *Proceedings of the International Conference on Recent advances in Space Technologies*, Istanbul, Turkey (Jun. 2009) pp. 710–715.
23. T. Miyabe, A. Konno and M. Uchiyama, "Automated Object Capturing with a Two-arm Flexible Manipulator," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan China (May 2003) vol. 2, pp. 2529–2534.
24. R. Fleurmond and V. Cadenat, "Multi-cameras Visual Servoing to Perform a Coordinated Task Using a Dual Arm Robot," *Proceedings of the International Conference on Informatics in control, automation and Robotics (ICINCO)*, Vienna, Austria (2014) vol. 2, pp. 37–44.
25. S. Huang, K. Murakami, Y. Yamakawa, T. Senoo and M. Ishikawa, "Fast Peg-and-hole Alignment Using Visual Compliance," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan (Nov. 2013) pp. 286–292.
26. H. W. Kim, J. S. Cho and I. S. Kweon, "A Novel Image-Based Control Law for the Visual Servoing System Under Large Pose Error," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Takamatsu, Japan (Oct. 2000) vol. 1, pp. 263–268.
27. O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Trans. Robot.* **30**(1), 244–257 (Feb. 2014).
28. F. Schramm, F. Geffard, G. Morel and A. Micaelli, "Calibration Free Image Point Path Planning Simultaneously Ensuring Visibility and Controlling Camera Path," *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Roma, Italy (2007) pp. 2074–2079.
29. F. Schramm, A. Micaelli and G. Morel, "Calibration Free Path Planning for Visual Servoing Yielding Straight Line Behavior Both in Image and Work Space," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Canada (Oct. 2005) vol. 1, pp. 2216–2221.
30. N. Mansard and F. Chaumette, "Tasking sequencing for high-level sensor-based control," *IEEE Trans. Robot.* **23**(1), 60–72 (Feb. 2007).

https://doi.org/10.1017/S0263574716000849 Published online by Cambridge University Press