Uncalibrated Path Planning in the Image Space for the Fixed Camera Configuration

LIANG Xin-Wu[1] HUANG Xin-Han[2] WANG Min[2]

Abstract Image-based visual servoing can be used to efficiently control the motion of robot manipulators. When the initial and the desired configurations are distant, however, as pointed out by many researchers, such a control approach can suffer from the convergence and stability problems due to its local properties. By specifying adequate image feature trajectories to be followed in the image, we can take advantage of the local convergence and stability of image-based visual servoing to avoid these problems. Hence, path planning in the image space has been an active research topic in robotics in recent years. However, almost all of the related results are established for the case of camera-in-hand configuration. In this paper, we propose an uncalibrated visual path planning algorithm for the case of fixed-camera configuration. This algorithm computes the trajectories of image features directly in the projective space such that they are compatible with rigid body motion. By decomposing the projective representations of the rotation and the translation into their respective canonical forms, we can easily interpolate their paths in the projective space. Then, the trajectories of image features in the image plane can be generated via projective paths. In this way, the knowledge of feature point structures and camera intrinsic parameters are not required. To validate the feasibility and performance of the proposed algorithm, simulation results based on the puma560 robot manipulator are given in this paper.

Key words Visual servoing, path planning, uncalibrated camera, projective space, fixed-camera configuration

Citation Liang Xin-Wu, Huang Xin-Han, Wang Min. Uncalibrated path planning in the image space for the fixed camera configuration. *Acta Automatica Sinica*, 2013, **39**(6): 759−769

DOI 10.3724/SP.J.1004.2013.00759

Traditional image-based visual servoing is usually based on a point-to-point control strategy, i.e., an error obtained by comparing the image features in the current image and in the constant desired one should be regulated to zero by some appropriate control laws. When the initial pose of the end-effector/camera is far away from its desired one, as pointed out by many researchers, the trajectories of the end-effector/camera induced by this strategy may be neither physically valid nor optimal such that the constraints of joint limits or field-of-view limits could be violated, which can lead to the task failure. By specifying adequate trajectories of image features in the image plane to be followed, we can fully utilize the local convergence and stability properties of image-based visual servoing to successfully accomplish the servoing task since in such a manner, the error to be regulated in each control cycle can be made small enough to be easily handled. Therefore, during the last decades, significant efforts have been devoted to visual path planning algorithms for generating trajectories of image features in the image space.

The difficulties existed in visual path planning lie in the fact that the generated image trajectories must correspond to physically valid relative-pose trajectories between the camera and the observed target. When the camera model and the 3D target model are assumed to be known, we can first obtain the initial and desired relative poses through 3D reconstruction in Euclidian space, respectively from the initial and desired image features. Then, the Euclidian trajectories can be derived from the interpolation of these two relative poses, from which we can easily deduce the feasible feature trajectories in the image space by using the perspective projection model of the camera. But this approach may require tedious camera calibrations and can only apply to some very limited environments. Hence, most of the works in visual path planning have been focused on the uncalibrated methods in order to eliminate the knowledge requirements of the exact camera intrinsic parameters and the target model. Without the use of the target model, the scaled Euclidian reconstruction approach can be adopted to obtain the rotation and the scaled translation motion for interpolation of the camera path between the initial and desired positions. The algorithms proposed in [1−10] are based on such a strategy. In [1], an optimal camera path planning algorithm with minimum acceleration was derived as a closed-form smooth collineation path. Besides the optimization criterion with respect to minimum acceleration, the minimum velocity problem was also considered[2]. To take into account the constraints encountered by the robot manipulator in the servoing process, a potential field-based method[3] was proposed to tackle both the robot joint-limits and the visibility-limits constraints. Similarly, self-occlusions and visibility-limits constraints were also addressed[4]. In order to handle the very large camera displacements and more complex scenes, image interpolation among multiple images was developed[5]. In [6], instead of a straight line interpolation for the camera translation motion, a generic helix trajectory was used to join the initial and desired positions of the camera in order to perfectly harmonize the geodesic trajectory of the rotation. To handle more general constraints and achieve better performance, a global optimization-based framework was proposed[7] by accounting for the visibility, workspace

Manuscript received January 21, 2011; accepted April 20, 2012
Supported by National Natural Science Foundation of China (60873032, 61105095, 61203361), Doctoral Program Foundation of Ministry of Education of China (20100142110020), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20100073120020), Postdoctoral Science Foundation of China (2012M511095), Shanghai Municipal Natural Science Foundation (11ZR1418400), and Shanghai Postdoctoral Science Foundation (12R21414200)
Recommended by Associate Editor ZHOU Jie
1. Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing of Ministry of Education of China, Shanghai 200240, China
2. Department of Control Science and Engineering, Huazhong University of Science and Technology, Key Laboratory of Image Processing and Intelligent Control of Ministry of Education of China, Wuhan 430074, China

and joint constraints while minimizing a cost function such as spanned image area, trajectory length and curvature. Note also that robust object reconstruction was used to cope with calibration errors and image noises and an extended Euler representation was adopted to parameterize the rotation matrix such that polynomial-optimization algorithms can be easily applied. Similarly, Chesi et al.[8] parameters were used to parameterize the rotation matrix such that a polynomial-optimization formulation can also be derived for keeping all features in the camera field of view. To deal with the local optimal solution problems in [7−8], a linear matrix inequality (LMI) optimization framework was developed[9] by using homogeneous forms to parameterize the camera trajectories and formulate various constraints and cost functions to be optimized. By making use of the powerful capabilities of the probabilistic roadmap approach, a sampling-based path planning algorithm was presented[10]. Once the camera trajectories are obtained in the scaled Euclidian space, the collineation path can be derived to map the initial image features to the current reference image features such that the image trajectories in the image space can hence be calculated. Another solution for getting the image trajectories is to directly map the scaled reconstructed initial positions of features onto the image plane by the use of the scaled Euclidian path as adopted in [7−9].

Note that the above-mentioned approaches depend on the knowledge of the camera intrinsic parameters more or less. To completely remove the dependence on the camera intrinsic parameters and the target model, path planning can be directly performed in the projective space, such as the approaches used in [11−14]. The traditional idea used in this strategy is to plan the rotation with a geodesic path while the translation path is interpolated by a straight line. But in that case the camera may face the field of view constraint problem. To solve such a problem, a new strategy called depth modulation was proposed[11−12] to ensure that all features were kept in the camera field of view. In [13], a translation modulation with three degrees of freedom was determined in the affine space by using geometric computations to ensure visibility of the whole target. To yield straight line behavior both in image and work space, a rotation modulation strategy was presented[14]. In the projective space-based approaches, the feature trajectories in the image space can be obtained by associating the planned projective path with the initial image features. Note that the previous methods are only valid when the cameras used for both learning and servoing tasks are exactly the same. To relax this condition, a path planning algorithm in the invariant projective space was proposed[15] to yield straight line behavior of the camera such that camera intrinsic parameters are not constrained to be the same.

It should be pointed out that the visual path planning approaches discussed so far are proposed for the case of camera-in-hand configuration. Contrarily, few results were devoted to the visual path planning problem for the fixed camera configuration. Using a fixed stereo rig, in [16−17], the path planning was completed in the projective space by screw decompositions of the conjugate transformation between the initial and desired projective coordinates of the gripper points, with the help of projective invariants. In order to avoid the mechanical constraints such as robot joint limits, an orientation-generating operator was proposed[18] in the projective space at the expense of an additional view of the gripper. This approach can provide three degrees of freedom for generating the orientation of the gripper.

It should be pointed out that the previous works mentioned above cannot be directly applied to solve the problem of image-space path planning with a single, uncalibrated and fixed camera. Compared with the case of image-space path planning with a single, uncalibrated and eye-in-hand camera, the main difficulty for this research is that, besides the unknown camera intrinsic parameters and the unknown relative motion between the initial and the desired position of the end-effector, we should also deal with an additional unknown relative pose between the camera frame and the initial end-effector frame in the case of image-space path planning with a single, uncalibrated and fixed camera, which can make the motion structure in the projective space more complicated and so can make the problem hard to solve. Then, how to solve this problem is the main objective of the research presented in this paper, i.e., in this paper, we will propose a visual path planning algorithm for the fixed-camera configuration, with a single uncalibrated camera observing the motion of the robot manipulator. In the learning stage, let the robot manipulator move to its desired position and the fixed camera is used to capture the corresponding image at this relative position, which is denoted as the desired image. Then, when the robot manipulator starts at an arbitrary initial position, we can also capture the image of the robot end-effector at this relative position, which is denoted as the initial image. With the initial and desired images at hand, the task of visual path planning is to generate the image feature trajectories in the image space that interpolate the initial and desired image features. The proposed visual path planning approach in this paper is based on interpolation in the projective space, and hence, it is a model-free approach such that the camera intrinsic parameters and the model of the target are not needed. To perform interpolation in the projective space, the projective representation between the initial and desired images should first be derived. It will be found that two additional movements of the robot manipulator from the initial position are required to recover the projective parameters, whereas only one additional movement was required in the case of camera-in-hand configuration[11] and no additional movement was needed in the case of fixed-camera configuration with a stereo rig[16−18]. But this is not a problem since the additional movements can easily be carried out automatically by the robot manipulator. Using the obtained projective parameters, we simply interpolate the rotation with a geodesic path and the translation with a straight line path, from which we can obtain the image feature trajectories to be followed by applying an image-based visual servoing controller. To show the validity and performance of the proposed approach, simulation results based on the PUMA560 robot manipulator are also given in this paper.

1 Modeling

Notations. Denote the relative pose of coordinate frame j with respect to coordinate frame i by the rotation ${}^iR_j$ and the translation ${}^i\boldsymbol{t}_j$, and the relative velocity of coordinate frame j with respect to coordinate frame i by the rotational velocity ${}^i\boldsymbol{\Omega}_j$ and the translational velocity ${}^i\boldsymbol{V}_j$, $i, j = b, c, 0, e, 1$, where b, c, 0, e, and 1 represent the robot base frame B, the camera frame C, the initial end-effector frame E^0, the current end-effector frame E, and the desired end-effector frame E^1, respectively, and the setup for the three coordinate frames E^0, E^1, and C is shown in Fig. 1. To simplify the notation, we use ${}^0R_1 := R$ and ${}^0\boldsymbol{t}_1 := \boldsymbol{t}$. Note that the homogeneous transformation ${}^iT_j$ is also used to represent the relative pose between coordinate frames i and j, which is composed of ${}^iR_j$ and ${}^i\boldsymbol{t}_j$.

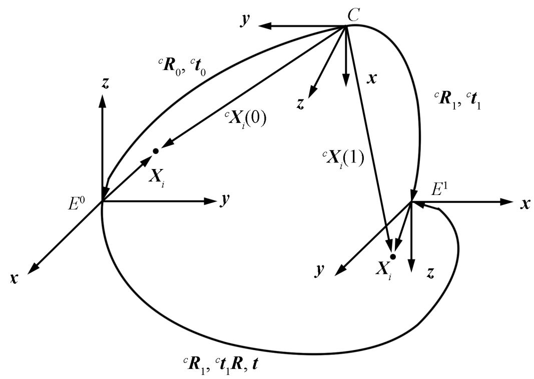


Fig. 1 Relationship among different coordinate frames in the modeling process

Let us assume that there are N feature points rigidly attached to the robot end-effector and Cartesian coordinates of the ith feature point are denoted by $\boldsymbol{X}_i$ with respect to the end-effector frame, which is constant. By applying coordinate transformations, when the end-effector is located at its initial position, Cartesian coordinates of the ith feature point with respect to the camera frame ${}^c\boldsymbol{X}_i(0)$ are given by

$$ {}^c\boldsymbol{X}_i(0) = {}^c\boldsymbol{t}_0 + {}^cR_0\boldsymbol{X}_i \tag{1} $$

In the same way, when the end-effector is located at its desired position, Cartesian coordinates of the i-th feature point with respect to the camera frame ${}^c\boldsymbol{X}_i(1)$ is given by

$$ {}^c\boldsymbol{X}_i(1) = {}^c\boldsymbol{t}_1 + {}^cR_1\boldsymbol{X}_i \tag{2} $$

Substituting (1) into (2) by eliminating $\boldsymbol{X}_i$, we can obtain the relationship between ${}^c\boldsymbol{X}_i(1)$ and ${}^c\boldsymbol{X}_i(0)$ as follows:

$$ {}^c\boldsymbol{X}_i(1) = {}^c\boldsymbol{t}_1 + {}^cR_1{}^cR_0^{-1}({}^c\boldsymbol{X}_i(0) - {}^c\boldsymbol{t}_0) \tag{3} $$

Based on the configuration as shown in Fig. 1, we have

$$ {}^cR_1 = {}^cR_0R \tag{4} $$

Substituting (4) into (3) yields

$$ {}^c\boldsymbol{X}_i(1) = {}^c\boldsymbol{t}_1 + {}^cR_0R{}^cR_0^{-1}({}^c\boldsymbol{X}_i(0) - {}^c\boldsymbol{t}_0) \tag{5} $$

It is noted that

$$ {}^c\boldsymbol{t}_1 = {}^cR_0\boldsymbol{t} + {}^c\boldsymbol{t}_0 \tag{6} $$

Substituting (6) into (5) leads to

$$ {}^c\boldsymbol{X}_i(1) = {}^cR_0R{}^cR_0^{-1}({}^c\boldsymbol{X}_i(0) - {}^c\boldsymbol{t}_0) + {}^c\boldsymbol{t}_0 + {}^cR_0\boldsymbol{t} \tag{7} $$

Under the perspective projection model of the camera, the point in the Euclidian space ${}^c\boldsymbol{X}_i(1) = [x_i^1, y_i^1, z_i^1]^{\mathrm{T}}$ is mapped into a point in the image space, which is denoted by $\boldsymbol{p}_i^1 = [u_i^1, v_i^1, 1]^{\mathrm{T}}$ in pixels as follows:

$$ z_i^1\boldsymbol{p}_i^1 = K{}^c\boldsymbol{X}_i(1) \tag{8} $$

where K denotes the matrix taking the camera intrinsic parameters[15]. Similarly, the point ${}^c\boldsymbol{X}_i(0) = [x_i^0, y_i^0, z_i^0]^{\mathrm{T}}$ is mapped into $\boldsymbol{p}_i^0 = [u_i^0, v_i^0, 1]^{\mathrm{T}}$ as follows:

$$ z_i^0\boldsymbol{p}_i^0 = K{}^c\boldsymbol{X}_i(0) \tag{9} $$

From (7) ∼ (9), we can easily obtain

$$ z_i^1\boldsymbol{p}_i^1 = G_0^1(z_i^0\boldsymbol{p}_i^0 - \boldsymbol{g}_0) + \boldsymbol{g}_0 + \boldsymbol{g}_0^1 \tag{10} $$

where

$$ G_0^1 = K{}^cR_0R{}^cR_0^{-1}K^{-1} \tag{11} $$

$$ \boldsymbol{g}_0 = z_0\boldsymbol{p}_0 = K{}^c\boldsymbol{t}_0 \tag{12} $$

$$ \boldsymbol{g}_0^1 = z_0^1\boldsymbol{p}_0^1 = K{}^cR_0\boldsymbol{t} \tag{13} $$

Note that z_0 and $\boldsymbol{p}_0$ are respectively the depth and the virtual image coordinates of the origin of coordinate frame E^0; $\boldsymbol{p}_0^1$ can be considered as a virtual epipole viewed in the camera frame.

Similarly to the statement given in [13], (10) can also be considered as generalizing the rigid body constraint to an affine 3D space. In the case of camera-in-hand configuration, the motion structure between the initial and desired camera positions is encoded by two parts[13], i.e., $G_{0\to1}$ and $\boldsymbol{g}_{0\to1}$. In the case of fixed-camera configuration considered in this paper, on the other hand, it is encoded by three parts, i.e., the projective rotation G_0^1, the virtual image point $\boldsymbol{g}_0$, and the virtual epipole $\boldsymbol{g}_0^1$.

2 Visual path planning algorithm

Note that the parameter $\boldsymbol{g}_0$ contained in (10) is constant during the motion of the end-effector when the initial end-effector position is regarded as a reference point. Hence, in order to generate an interpolated path, trajectories G_0^σ and $\boldsymbol{g}_0^\sigma$ should be provided such that the trajectories $z_i^\sigma\boldsymbol{p}_i^\sigma$ for any feature point $\boldsymbol{X}_i$ can be respectively derived as follows:

$$ z_i^\sigma\boldsymbol{p}_i^\sigma = G_0^\sigma(z_i^0\boldsymbol{p}_i^0 - \boldsymbol{g}_0) + \boldsymbol{g}_0 + \boldsymbol{g}_0^\sigma \tag{14} $$

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

$$ G_0^\sigma|_{\sigma=0} = I_3\boldsymbol{g}_0^\sigma|_{\sigma=0} = \boldsymbol{0} \tag{15} $$

$$ G_0^\sigma|_{\sigma=1} = G_0^1\boldsymbol{g}_0^\sigma|_{\sigma=1} = \boldsymbol{g}_0^1 \tag{16} $$

To simplify the visual path planning problem, linear interpolation will be used for generating the trajectories, i.e.

$$ \boldsymbol{g}_0^\sigma = \sigma\boldsymbol{g}_0^1\sigma \in [0, 1] \tag{17} $$

To generate the image feature trajectories using (14), the projective motion parameters G_0^1 and $\boldsymbol{g}_0^1$, and the virtual image feature $\boldsymbol{g}_0$ should be first recovered.

2.1 Reconstruction of depth fields z_i^0 by a pure translation motion from the initial end-effector position

In the case of camera-in-hand configuration, the depth fields can be recovered by a pure translation motion from the initial camera position[11]. It is still effective in the case of fixed-camera configuration, as shown in the following.

Let the robot end-effector move from its initial pose by a pure translation motion such that $R = I_3$ and $\boldsymbol{t} = \bar{\boldsymbol{t}}$. According to (10), (11) and (13), we have

$$z_i^t \boldsymbol{p}_i^t = z_i^0 \boldsymbol{p}_i^0 + \boldsymbol{g}_0^t \tag{18}$$

where $\boldsymbol{g}_0^t = K^c R_0 \bar{\boldsymbol{t}}$ and $\boldsymbol{p}_i^t$ is the image coordinates of the i-th feature point when the end-effector is located at the new position after the translation motion. Then, for another feature point m, we also have

$$z_m^t \boldsymbol{p}_m^t = z_m^0 \boldsymbol{p}_m^0 + \boldsymbol{g}_0^t \tag{19}$$

Subtracting (19) from (18), we obtain

$$z_i^t \boldsymbol{p}_i^t - z_m^t \boldsymbol{p}_m^t = z_i^0 \boldsymbol{p}_i^0 - z_m^0 \boldsymbol{p}_m^0 \tag{20}$$

Combining N feature points together and using the same notations as adopted in [11], we can easily have

$$\left[B(\boldsymbol{p}^t) - B(\boldsymbol{p}^0) \right] \left[\begin{array}{c} \boldsymbol{Z}^t \\ \boldsymbol{Z}^0 \end{array} \right] = \boldsymbol{0} \tag{21}$$

where $\boldsymbol{Z}^t$ and $\boldsymbol{Z}^0$ are the depth fields defined by $\boldsymbol{Z}^t = [z_1^t, \cdots, z_N^t]^{\mathrm{T}}$ and $\boldsymbol{Z}^0 = [z_1^0, \cdots, z_N^0]^{\mathrm{T}}$, and $B(\boldsymbol{p})$ is given by

$$B(\boldsymbol{p}) = \left[\begin{array}{ccccc} \boldsymbol{p}_1 & -\boldsymbol{p}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{p}_2 & -\boldsymbol{p}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{p}_{N-1} & -\boldsymbol{p}_N \\ -\boldsymbol{p}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{p}_N \end{array} \right]$$

As stated in [11], at least $N = 3$ matched points can be used to solve the linear homogenous system up to a common scalar factor. In other words, the solution can be obtained by solving the general linear homogeneous problem using the algorithm of singular value decomposition (SVD). Assuming that $\mathrm{SVD}\left\{ \left[B(\boldsymbol{p}^t) - B(\boldsymbol{p}^0) \right] \right\} = \left[U_t, S_t, V_t \right]$, where SVD denotes the operator of singular value decomposition, and then, the solution $\left[\boldsymbol{Z}^{t\mathrm{T}}, \boldsymbol{Z}^{0\mathrm{T}} \right]^{\mathrm{T}}$ can be taken as the last column of V_t, which corresponds to the singular value 0 in the absence of noises, or corresponds to the smallest singular value in the presence of noises.

2.2 Reconstruction of the virtual image point $\boldsymbol{g}_0$ by a pure rotational motion from the initial end-effector position

After the depth fields have been found, in the case of camera-in-hand configuration, the other projective motion parameters can be solved from the corresponding points between the initial and desired image features. In the case of fixed-camera configuration, however, another pure rotation motion is needed to obtain the virtual image point $\boldsymbol{g}_0$.

Let the robot end-effector move from its initial pose by a pure rotation such that $R = \bar{R}$ and $\boldsymbol{t} = 0$. According to (10), (11) and (13), we have

$$z_i^r \boldsymbol{p}_i^r = G_0^r (z_i^0 \boldsymbol{p}_i^0 - \boldsymbol{g}_0) + \boldsymbol{g}_0 \tag{22}$$

where $G_0^r = K^c R_0 \bar{R}^c R_0^{-1} K^{-1}$ and $\boldsymbol{p}_i^r$ is the image coordinates of the i-th feature point when the end-effector is located at the new position after the rotation motion. Let $\bar{\boldsymbol{g}} = (I - G_0^r)\boldsymbol{g}_0$, we have

$$z_i^r \boldsymbol{p}_i^r = G_0^r z_i^0 \boldsymbol{p}_i^0 + \bar{\boldsymbol{g}} \tag{23}$$

since $z_i^0 \boldsymbol{p}_i^0$ is a known quantity, by stacking z_i^r, $\bar{\boldsymbol{g}}$ and the elements of G_0^r into a solution vector as the case for the depth fields in (21), we can also obtain their values up to a common scalar factor by employing SVD. Hence, $\bar{\boldsymbol{g}}$ and G_0^r can be deduced from the solution vector. Then, by solving the linear system $(I - G_0^r)\boldsymbol{g}_0 = \bar{\boldsymbol{g}}$, we can derive the virtual image point $\boldsymbol{g}_0$.

It should be pointed out that different from the method used to obtain the solution in (21), the solution vector here cannot be simply taken as the last column of V_r, where we assume that three matrices U_r, S_r and V_r are obtained by applying SVD. Specifically, it should be taken as an appropriately normalized vector of the last column of V_r in order to obtain the matrix G_0^r with an eigenvalue of one such that this matrix actually corresponds to a projective rotation matrix, since G_0^r is similar to the rotation matrix $\bar{R}$ and it must have an eigenvalue of one. This is of great importance to the success of computing the solution of $\boldsymbol{g}_0$. Suppose that the last column of V_r is denoted by $\boldsymbol{\xi}_1$. From $\boldsymbol{\xi}_1$, we can obtain $\bar{G}_0^r$, which is different from G_0^r up to a common scalar factor. We assume that the eigenvalues of $\bar{G}_0^r$ are given by α_1, α_2 and α_2^*, where $*$ denotes the conjugate operator. Then, the solution vector $\boldsymbol{\xi}$ can be taken as $\boldsymbol{\xi} = \boldsymbol{\xi}_1/\alpha_1$. Now, $\bar{\boldsymbol{g}}$ and G_0^r can be derived from the solution vector $\boldsymbol{\xi}$ such that one of the eigenvalues of G_0^r is one in the ideal condition, which is a necessary property of this matrix.

After having obtained $\bar{\boldsymbol{g}}$ and G_0^r, more attention should be paid to solving the linear system $(I - G_0^r)\boldsymbol{g}_0 = \bar{\boldsymbol{g}}$ to obtain $\boldsymbol{g}_0$. To obtain a stable solution, the SVD approach should be used to solve this system, especially in the presence of noises or numerical errors. Suppose that $\mathrm{SVD}\left\{ I - G_0^r \right\} = \left[U_{r0}, S_{r0}, V_{r0} \right]$, we have $U_{r0} S_{r0} V_{r0}^{\mathrm{T}} \boldsymbol{g}_0 = \bar{\boldsymbol{g}}$. Let $\bar{\boldsymbol{g}}_0 = V_{r0}^{\mathrm{T}} \boldsymbol{g}_0$ and $\bar{\bar{\boldsymbol{g}}} = U_{r0}^{\mathrm{T}} \bar{\boldsymbol{g}}$, we obtain $S_{r0} \bar{\boldsymbol{g}}_0 = \bar{\bar{\boldsymbol{g}}}$, from which we can easily derive the robust solution of $\bar{\boldsymbol{g}}_0$ as given by $\bar{\boldsymbol{g}}_0 = [(\bar{S}_{r0}^{-1} \bar{\bar{\boldsymbol{g}}})^{\mathrm{T}}, 0]^{\mathrm{T}} + \kappa [0, 0, 1]^{\mathrm{T}}$, where $\bar{S}_{r0}$ is the leading 2×2 submatrix of S_{r0}, $\bar{\bar{\boldsymbol{g}}}$ is the leading 2×1 subvector of $\bar{\bar{\boldsymbol{g}}}$ and κ is an arbitrary constant. Then, the solution to the virtual image $\boldsymbol{g}_0$ is given by $\boldsymbol{g}_0 = V_{r0} \bar{\boldsymbol{g}}_0$.

2.3 Reconstruction of projective motion parameters G_0^1 and g_0^1 from matched points between the initial and desired end-effector positions

Now, (10) can be rewritten as follows:

$$z_i^1 \boldsymbol{p}_i^1 = G_0^1 \bar{\boldsymbol{g}}_{i,0} + \bar{\boldsymbol{g}}_0^1 \tag{24}$$

where $\bar{\boldsymbol{g}}_{i,0} = z_i^0 \boldsymbol{p}_i^0 - \boldsymbol{g}_0$ and $\bar{\boldsymbol{g}}_0^1 = \boldsymbol{g}_0 + \boldsymbol{g}_0^1$. Since the depth fields z_i^0 and the virtual image point $\boldsymbol{g}_0$ are already known, $\bar{\boldsymbol{g}}_{i,0}$ is a known quantity. Hence, similar to the previous developments, we can easily obtain the parameters z_i^1, G_0^1 and

$\bar{\boldsymbol{g}}_0^1$ by using the SVD approach. Then, the virtual epipole $\boldsymbol{g}_0^1$ can be easily derived from $\bar{\boldsymbol{g}}_0^1 = \boldsymbol{g}_0 + \boldsymbol{g}_0^1$.

Up to now, the projective motion parameters G_0^1 and $\boldsymbol{g}_0^1$, and the virtual image point $\boldsymbol{g}_0$ presented in (10) all have been derived. Compared with the case of camera-in-hand configuration, an additional pure rotation motion is needed to recover these parameters and more calculations are required to solve related linear equations.

2.4 Path planning for the projective rotation motion G_0^σ in the projective space

Note that the collineation matrix given in (11) is slightly different from that in the case of camera-in-hand configuration, as we can see that an additional rotation matrix ${}^cR_0$ is included here. But this will not bring any big problem since this matrix is constant. Hence, interpolation of G_0^1 also comes down to the interpolation of R. As pointed out in [13], the rotation matrix R can be represented via eigen-decomposition, in terms of rotation angle θ in the matrix Λ of eigenvalues and rotation axes $\boldsymbol{u}_1$ in the matrix U as follows:

$$R = U\Lambda(\theta)U^* \tag{25}$$

with $\Lambda(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{e}^{j\theta} & 0 \\ 0 & 0 & \mathrm{e}^{-j\theta} \end{bmatrix}$ and $U = [\boldsymbol{u}_1\ \boldsymbol{u}_2\ \boldsymbol{u}_2^*]$. When R, K and ${}^cR_0$ are all known, we can first interpolate the eigen-matrix $\Lambda(\theta)$ as $\Lambda(\sigma\theta)$, $\sigma \in [0, 1]$, and then, the interpolation of G_0^1 can be calculated as follows:

$$G_0^\sigma = K{}^cR_0U\Lambda(\sigma\theta)U^*{}^cR_0^{-1}K^{-1} \tag{26}$$

But this is not the case since the matrices R, K and ${}^cR_0$ are not known to us. Fortunately, as the case for camera-in-hand configuration, the collineation matrix G_0^1 is still similar to the rotation matrix R. Hence, they have the same eigenvalues. In other words, the eigen-decomposition of G_0^1 shares the same eigen-matrix $\Lambda(\theta)$ as the rotation matrix R, i.e.

$$G_0^1 = V\Lambda(\theta)V^{-1} \tag{27}$$

where V represents the associated eigenvectors, which can be obtained from the knowledge of G_0^1. Compared with the structures of (26) and (27), we know that $V = K{}^cR_0U$. Though we cannot obtain the matrices U, K and ${}^cR_0$, we can know their product V through the eigen-decomposition of G_0^1. Hence, interpolation of G_0^1 can also be carried out by using (27) in this way:

$$G_0^\sigma = V\Lambda(\sigma\theta)V^{-1}\sigma \in [0, 1] \tag{28}$$

which provides the same solution as that of (26), but without any knowledge requirements of the matrices R, K and ${}^cR_0$.

2.5 Visual path planning for image feature points in the image space

Up to now, we have provided linear interpolation for the translation motion as shown in (17) and the geodesic interpolation for the rotation motion as shown in (28). Therefore, the image trajectories of the i-th feature point can be generated via the use of (14), i.e., we can obtain the vector $\boldsymbol{w}_i^\sigma = z_i^\sigma \boldsymbol{p}_i^\sigma$, $\sigma \in [0, 1]$. Then, we can derive the image coordinates of the i-th feature point in the following way:

$$u_i^\sigma = \frac{w_{i,1}^\sigma}{w_{i,3}^\sigma},\ v_i^\sigma = \frac{w_{i,2}^\sigma}{w_{i,3}^\sigma} \tag{29}$$

where $w_{i,j}^\sigma$ denotes the j-th component of $\boldsymbol{w}_i^\sigma$. Then, $\boldsymbol{p}_i^\sigma$ is given by $\boldsymbol{p}_i^\sigma = [u_i^\sigma, v_i^\sigma, 1]^{\mathrm{T}}$ at the time instant σ. In this way, the obtained image trajectories of feature points in the image space correspond to physically valid rigid motion of the robot end-effector and the generated trajectories can be tracked using image-based visual servoing techniques in order to fully take advantage of the local convergence and robustness of image-based visual servoing methods.

3 Image-based visual servoing for the fixed-camera configuration

For the purpose of simpleness, we only consider the kinematics-based visual servoing approach. Specifically speaking, the robot joint velocity $\dot{\boldsymbol{q}}(\tau) \in \mathbf{R}^{n\times 1}$ is taken as the control input needed to be designed. In other words, the joint velocity can be exactly generated to control the motion of the robot manipulator immediately after the command of the kinematics-based controller is issued.

Differentiating the image formation equation $z_i^\tau \boldsymbol{p}_i^\tau = K{}^c\boldsymbol{X}_i(\tau)$, we obtain

$$\dot{z}_i^\tau \boldsymbol{p}_i^\tau + z_i^\tau \dot{\boldsymbol{p}}_i^\tau = K{}^c\dot{\boldsymbol{X}}_i(\tau) \tag{30}$$

Considering first the component u_i^τ of $\boldsymbol{p}_i^\tau$, we have

$$z_i^\tau \dot{u}_i^\tau = \boldsymbol{k}_1^{\mathrm{T}}{}^c\dot{\boldsymbol{X}}_i(\tau) - \dot{z}_i^\tau u_i^\tau \tag{31}$$

Using $\dot{z}_i^\tau = \boldsymbol{k}_3^{\mathrm{T}}{}^c\dot{\boldsymbol{X}}_i(\tau)$ results in

$$\dot{u}_i^\tau = \frac{1}{z_i^\tau}(\boldsymbol{k}_1^{\mathrm{T}} - u_i^\tau \boldsymbol{k}_3^{\mathrm{T}}){}^c\dot{\boldsymbol{X}}_i(\tau) \tag{32}$$

where $\boldsymbol{k}_i^{\mathrm{T}}$ is the i-th row of K. It is noted that[19]

$${}^c\dot{\boldsymbol{X}}_i(\tau) = {}^c\boldsymbol{\Omega}_e \times {}^c\boldsymbol{X}_i(\tau) + {}^c\boldsymbol{V}_e \tag{33}$$

Substituting ${}^c\boldsymbol{X}_i(\tau) = z_i^\tau K^{-1}\boldsymbol{p}_i^\tau$ into (33) yields

$${}^c\dot{\boldsymbol{X}}_i(\tau) = \left[I_3 - z_i^\tau \left[K^{-1}\boldsymbol{p}_i^\tau\right]_\times\right] \begin{bmatrix} {}^c\boldsymbol{V}_e \\ {}^c\boldsymbol{\Omega}_e \end{bmatrix} \tag{34}$$

where $[\boldsymbol{a}]_\times$ denotes a skew-symmetric matrix defined by

$$[\boldsymbol{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

with $\boldsymbol{a} = [a_1, a_2, a_3]^{\mathrm{T}}$. Combining (32) and (34), we have

$$\dot{u}_i^\tau = J_{img,u_i} \begin{bmatrix} {}^c\boldsymbol{V}_e \\ {}^c\boldsymbol{\Omega}_e \end{bmatrix} \tag{35}$$

where

$$J_{img,u_i} = \begin{bmatrix} \frac{1}{z_i^\tau}(\boldsymbol{k}_1^{\mathrm{T}} - u_i^\tau \boldsymbol{k}_3^{\mathrm{T}}) & -(\boldsymbol{k}_1^{\mathrm{T}} - u_i^\tau \boldsymbol{k}_3^{\mathrm{T}})\left[K^{-1}\boldsymbol{p}_i^\tau\right]_\times \end{bmatrix}$$

According to [19], the velocities ${}^c\boldsymbol{V}_e$ and ${}^c\boldsymbol{\Omega}_e$ can be related to the velocities ${}^b\boldsymbol{V}_e$ and ${}^b\boldsymbol{\Omega}_e$ as follow:

$$\begin{bmatrix} {}^c\boldsymbol{V}_e \\ {}^c\boldsymbol{\Omega}_e \end{bmatrix} = J_{cst} \begin{bmatrix} {}^b\boldsymbol{V}_e \\ {}^b\boldsymbol{\Omega}_e \end{bmatrix} \tag{36}$$

where J_{cst} is defined by

$$J_{cst} = \left[\begin{array}{cc} {}^cR_b & [{}^c\boldsymbol{t}_b]_\times {}^cR_b \\ 0_{3\times 3} & {}^cR_b \end{array} \right]$$

Substituting (36) into (35) leads to

$$\dot{u}_i^\tau = J_{img,u_i} J_{cst} \left[\begin{array}{c} {}^b\boldsymbol{V}_e \\ {}^b\boldsymbol{\Omega}_e \end{array} \right] \tag{37}$$

It is also noted that

$$\left[\begin{array}{c} {}^b\boldsymbol{V}_e \\ {}^b\boldsymbol{\Omega}_e \end{array} \right] = J(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{38}$$

where $J(\boldsymbol{q})$ is the manipulator Jacobian matrix. Then, we can derive the velocity of the component u_i^τ induced by the robot joint velocity as follows:

$$\dot{u}_i^\tau = J_{u_i}(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q})\dot{\boldsymbol{q}} \tag{39}$$

where $J_{u_i}(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q})$ represents the overall image Jacobian matrix corresponding to the component u_i^τ:

$$J_{u_i}(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q}) = J_{img,u_i} J_{cst} J(\boldsymbol{q}) \tag{40}$$

Similarly, we can also derive the velocity of the component v_i^τ induced by the robot joint velocity as follows:

$$\dot{v}_i^\tau = J_{v_i}(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q})\dot{\boldsymbol{q}} \tag{41}$$

where $J_{v_i}(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q})$ denotes the overall image Jacobian matrix corresponding to the component v_i^τ:

$$J_{v_i}(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q}) = J_{img,v_i} J_{cst} J(\boldsymbol{q}) \tag{42}$$

with

$$J_{img,v_i} = \left[\begin{array}{cc} \frac{1}{z_i^\tau}(\boldsymbol{k}_1^{\rm T} - v_i^\tau \boldsymbol{k}_3^{\rm T}) & -(\boldsymbol{k}_1^{\rm T} - v_i^\tau \boldsymbol{k}_3^{\rm T})\Big[K^{-1}\boldsymbol{p}_i^\tau\Big]_\times \end{array} \right]$$

The image-plane velocity of the i-th feature point $\boldsymbol{f}_i^\tau = [u_i^\tau, v_i^\tau]^{\rm T}$ can then be given by

$$\dot{\boldsymbol{f}}_i^\tau = \left[\begin{array}{c} J_{u_i}(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q}) \\ J_{v_i}(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q}) \end{array} \right] \dot{\boldsymbol{q}} := J_i(\boldsymbol{p}_i^\tau, z_i^\tau, \boldsymbol{q})\dot{\boldsymbol{q}} \tag{43}$$

Combining all the N feature points together, we obtain

$$\dot{\boldsymbol{f}}^\tau = J(\boldsymbol{p}^\tau, \boldsymbol{z}^\tau, \boldsymbol{q})\dot{\boldsymbol{q}} \tag{44}$$

where $\boldsymbol{f}^\tau = \Big[(\boldsymbol{f}_1^\tau)^{\rm T}, \cdots, (\boldsymbol{f}_N^\tau)^{\rm T}\Big]^{\rm T}$, $\boldsymbol{z}^\tau = [z_1^\tau, \cdots, z_N^\tau]^{\rm T}$, $J(\boldsymbol{p}^\tau, \boldsymbol{z}^\tau, \boldsymbol{q}) = \Big[J_1^{\rm T}(\boldsymbol{p}_1^\tau, z_1^\tau, \boldsymbol{q}), \cdots, J_N^{\rm T}(\boldsymbol{p}_N^\tau, z_N^\tau, \boldsymbol{q})\Big]^{\rm T}$ is the overall image Jacobian matrix.

The objective of kinematics-based visual servoing for the fixed-camera configuration is to design appropriate control laws for generating the robot joint velocity $\dot{\boldsymbol{q}}$ to drive the image features $\boldsymbol{f}^\tau$ toward their goal position, which is specified by $\boldsymbol{f}_d$. The control law adopted in this paper for generating the control input $\dot{\boldsymbol{q}}$ is given as follows:

$$\dot{\boldsymbol{q}} = -J^+(\boldsymbol{p}^\tau, \boldsymbol{z}^\tau, \boldsymbol{q})\tilde{\boldsymbol{f}} \tag{45}$$

where $\tilde{\boldsymbol{f}} = \boldsymbol{f}^\tau - \boldsymbol{f}_d$ is the regulation error of feature points and $J^+(\boldsymbol{p}^\tau, \boldsymbol{z}^\tau, \boldsymbol{q})$ denotes an appropriate inverse of $J(\boldsymbol{p}^\tau, \boldsymbol{z}^\tau, \boldsymbol{q})$[20]. It can be easily shown that the regulation error $\tilde{\boldsymbol{f}}$ will be convergent to zero[21] under the condition that $\mathrm{rank}\Big(J(\boldsymbol{p}^\tau, \boldsymbol{z}^\tau, \boldsymbol{q})\Big) = \min(m, n)$ with $m = 2N$, i.e., the image features will be convergent to their desired values as time tends to infinity.

It should be pointed out that the Jacobian matrix $J(\boldsymbol{p}^\tau, \boldsymbol{z}^\tau, \boldsymbol{q})$ depends on the knowledge of the camera intrinsic and extrinsic parameters and the depths of the feature points, as we can see from both (40) and (42). Hence, when only approximate knowledge of these parameters is available, we can then obtain its approximate value $\hat{J}(\boldsymbol{p}^\tau, \hat{\boldsymbol{z}}^\tau, \boldsymbol{q}, \hat{K}, {}^c\hat{R}_b, {}^c\hat{\boldsymbol{t}}_b)$ by using the estimated values $\hat{\boldsymbol{z}}^\tau$, $\hat{K}$, ${}^c\hat{R}_b$ and ${}^c\hat{\boldsymbol{t}}_b$. In this case, the control law for generating the control input $\dot{\boldsymbol{q}}$ is given by

$$\dot{\boldsymbol{q}} = -\hat{J}^+(\boldsymbol{p}^\tau, \hat{\boldsymbol{z}}^\tau, \boldsymbol{q}, \hat{K}, {}^c\hat{R}_b, {}^c\hat{\boldsymbol{t}}_b)\tilde{\boldsymbol{f}} \tag{46}$$

where $\hat{J}^+(\boldsymbol{p}^\tau, \hat{\boldsymbol{z}}^\tau, \boldsymbol{q}, \hat{K}, {}^c\hat{R}_b, {}^c\hat{\boldsymbol{t}}_b)$ also represents an appropriate inverse of $\hat{J}(\boldsymbol{p}^\tau, \hat{\boldsymbol{z}}^\tau, \boldsymbol{q}, \hat{K}, {}^c\hat{R}_b, {}^c\hat{\boldsymbol{t}}_b)$.

4 Simulation results

To validate the proposed visual path planning algorithm, in this section, we will present simulation results by using a PUMA560 robot manipulator, which is based on the robotics toolbox for Matlab developed by Corke[22]. The end-effector of the robot manipulator is required to move to its desired position given by

$${}^bT_1 = \left[\begin{array}{cccc} -0.1194 & -0.4845 & -0.8666 & 0.3936 \\ 0.3010 & 0.8141 & -0.4966 & -0.7112 \\ 0.9461 & -0.3201 & 0.0486 & 0.0568 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

We assume that eight vertices of a hexahedron virtually attached to the robot end-effector are used as feature points to complete both the path planning and visual servoing tasks, whose Cartesian coordinates are given by

$$\begin{array}{l} \boldsymbol{X}_1 = [\chi, 0, 0]^{\rm T}, \boldsymbol{X}_2 = [\chi, \chi, 0]^{\rm T}, \boldsymbol{X}_3 = [\chi, \chi, \chi]^{\rm T} \\ \boldsymbol{X}_4 = [0, \chi, \chi]^{\rm T}, \boldsymbol{X}_5 = [0, \chi, 0]^{\rm T}, \boldsymbol{X}_6 = [0, 0, 0]^{\rm T} \\ \boldsymbol{X}_7 = [0, 0, \chi]^{\rm T}, \boldsymbol{X}_8 = [\chi, 0, \chi]^{\rm T} \end{array}$$

where $\chi = 0.08\,\mathrm{m}$. The initial robot end-effector position is assumed to be specified at the ready position of PUMA560, i.e.

$${}^bT_0 = \left[\begin{array}{cccc} 1.0000 & 0 & 0 & 0.0203 \\ 0 & 1.000 & 0 & -0.1500 \\ 0 & 0 & 1.000 & 0.8636 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

Note that a camera with intrinsic parameters:

$$K = \left[\begin{array}{ccc} 800 & 100 & 200 \\ 0 & 800 & 200 \\ 0 & 0 & 1 \end{array} \right]$$

is fixed near the robot manipulator to observe the motion of the end-effector. In addition, we assume that

$${}^b\boldsymbol{t}_c = [t_1, t_2, t_3]^{\rm T}, {}^bR_c = Rot(\boldsymbol{z}, \gamma)Rot(\boldsymbol{y}, \beta)Rot(\boldsymbol{x}, \alpha)$$

where $Rot(\iota, \phi)$ ($\iota = \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$) and $\phi = \alpha, \beta, \gamma$, respectively denotes a 3×3 basic rotation matrix about ι-axis by angle ϕ, $t_1 = -0.2056$, $t_2 = 1.1440$, $t_3 = -2.3236$, $\alpha = 2\pi/3$, $\beta = 2\pi/3$ and $\gamma = 2\pi/5$.

As our main concerns in this paper, we generate the intermediate image coordinates of the eight feature points, which are used to connect their initial images with the desired ones. Applying the proposed visual path planning algorithm yields the desired reference trajectories of the feature points in the image space, as shown in Fig. 2.

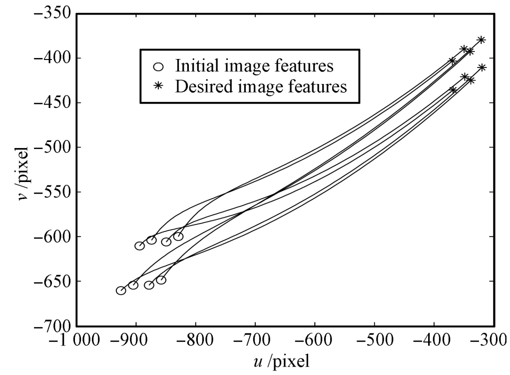
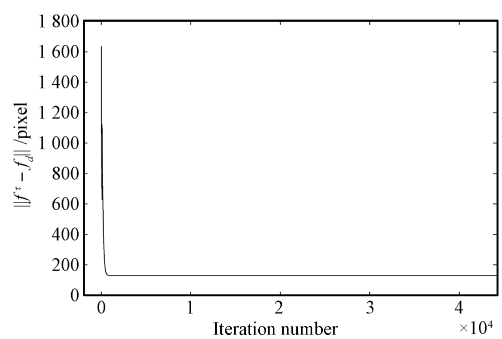
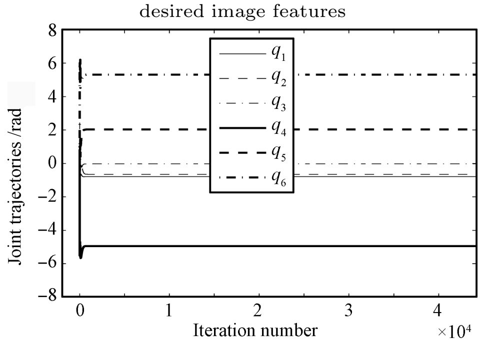


Fig. 2 Reference trajectories generated by the proposed algorithm

To show the necessity of the proposed algorithm, image-based visual servoing method presented in Section 3 is used to control the motion of the robot end-effector when the path planning algorithm is not used for generating intermediate reference image features. In other words, the initial image features captured at the initial position of the end-effector are required to be directly regulated to the constant desired ones captured at the desired position. In this case, the visual servoing approach fails to achieve such a goal and the corresponding results are given in Fig. 3, from which we can see that local minimum is reached and the robot joints violate their joint limits. Then, the initial image features cannot be regulated to their desired values and thus the desired position of the end-effector cannot be reached. On the other hand, when intermediate reference image features are generated to be followed by using the proposed algorithm, the initial image features can be successfully regulated to their corresponding desired values as illustrated in Figs. 4 ~ 6, from which we can know that the errors between the current and constant desired image features gradually decrease to zero when the iteration number increases and the ranges of the joint trajectories can be lowered by increasing intermediate points. Hence, when more intermediate reference image features are generated to be tracked, more optimal joint trajectories can be obtained and then the joint limits problem is expected to be solved. Continuing to increase intermediate points, the corresponding results are given in Fig. 7 for the case with 60 intermediate points, from which we know that the robot can successfully avoid its joint limits when the joint limits are considered as $(-\pi, \pi)$ for all joints. However, as the number of the intermediate points continues to increase, the ranges of the joint trajectories cannot be further lowered any more, as we can see from Fig. 8 for the case with 80 intermediate points. Hence, the joint limits problem can be solved to a certain extent by increasing intermediate points but it cannot be completely solved in this manner and deserves further investigation in the future research. For the purpose of clarity, we also provide the corresponding followed trajectories of image features in Fig. 9, from which we can see that the planned and followed trajectories are almost the same by comparing with Fig. 2. From the previous results, we can know that sufficient intermediate points should be

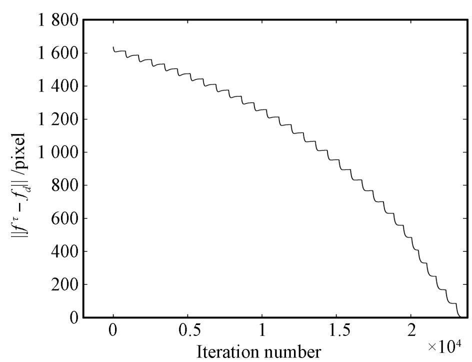


(a) Image feature error between the current and the constant desired image features

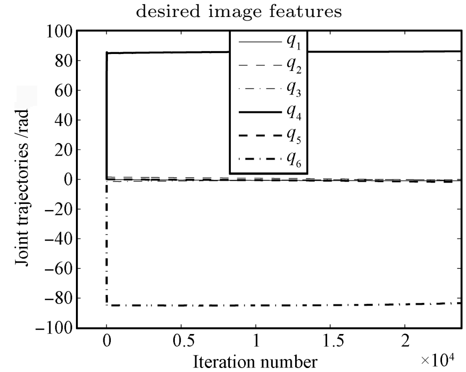


(b) Joint trajectories

Fig. 3 Visual servoing results for the case with no intermediate points

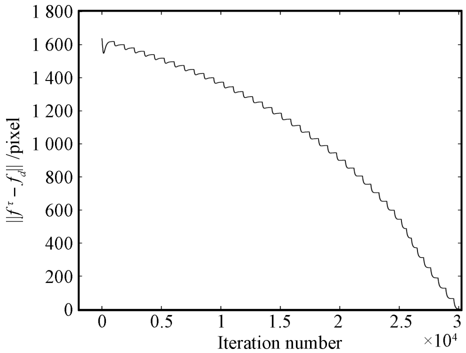


(a) Image feature error between the current and the constant desired image features

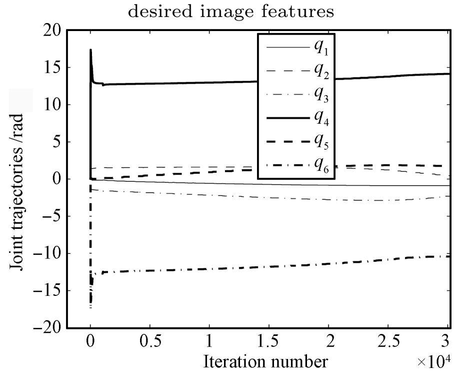


(b) Joint trajectories

Fig. 4 Visual servoing results for the case with 30 intermediate points

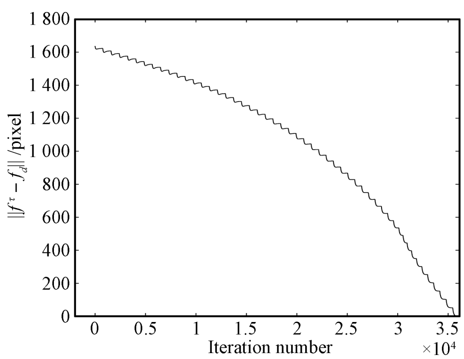


(a) Image feature error between the current and the constant desired image features

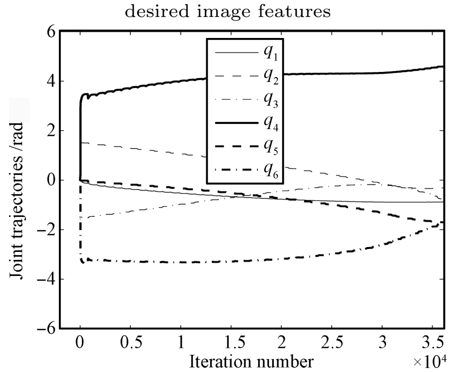


(b) Joint trajectories

Fig. 5 Visual servoing results for the case with 40 intermediate points

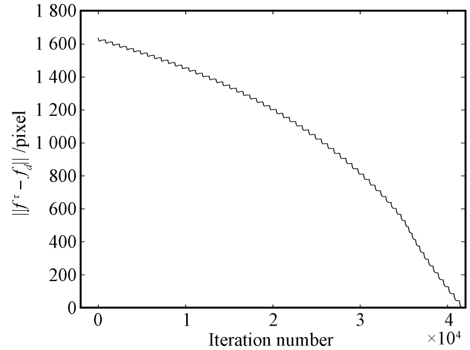


(a) Image feature error between the current and the constant desired image features

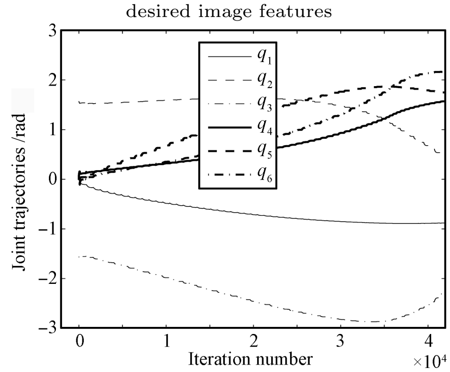


(b) Joint trajectories

Fig. 6 Visual servoing results for the case with 50 intermediate points

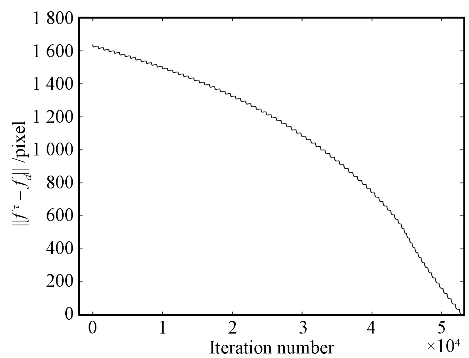


(a) Image feature error between the current and the constant desired image features

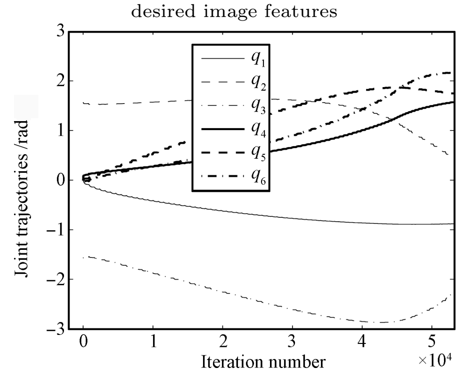


(b) Joint trajectories

Fig. 7 Visual servoing results for the case with 60 intermediate points



(a) Image feature error between the current and the constant desired image features



(b) Joint trajectories

Fig. 8 Visual servoing results for the case with 80 intermediate points

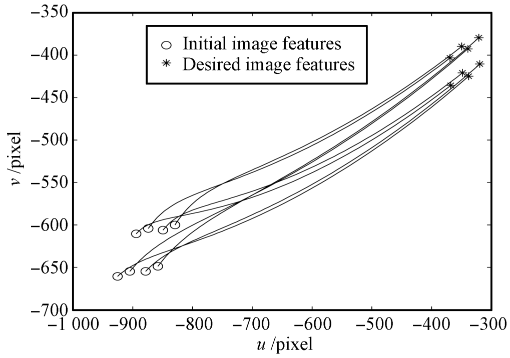


Fig. 9 Followed trajectories of the image features obtained by the visual servoing approach

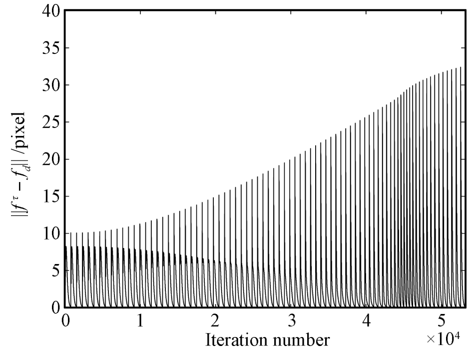
generated to obtain satisfying system performance.

As another main concern in this paper, we want to know whether the generated intermediate reference image features correspond to physically valid 3D rigid motion of the robot end-effector or not. In other words, it should be ensured that the image coordinates of the eight feature points can be made arbitrarily close to their respective reference positions at every intermediate reference image. This can be easily checked by using the visual servoing strategy adopted here: the visual servoing method used in this paper is still based on the point-to-point regulation strategy for simplicity, i.e., the intermediate reference image features nearest to the current image are taken as the current desired ones to be regulated and when the errors between them are below an arbitrarily specified small threshold, the next intermediate reference image features will be the next desired ones to be regulated. Then, these errors should be small enough to guarantee the feasibility of the generated intermediate reference image features. To show that this is certainly the case, simulation results about these errors are given in Fig. 10, from which we see that norms of these errors at the end of each regulation period are all below 0.1. Actually, arbitrarily small threshold can be attained but only more iterations are required. To understand that the end-effector is indeed reached to its desired pose with respect to the robot base frame when the image features are regulated to their desired position, the error trajectories of the end-effector between its current and desired poses are given in Fig. 11. Note that the actual trajectory of the robot end-effector in 3D space is also provided in Fig. 12. Then, it is found that the desired pose of the end-effector can be successfully attained using the proposed approach.

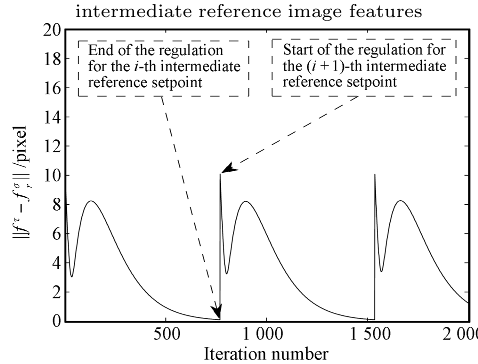
It should be pointed out that there exist unsmoothed changes in the simulation results. This is due to the switching of intermediate reference image features as the current desired ones to be regulated. To remove this phenomenon, smooth image feature trajectories should be derived by using natural cubic B-splines[4] to interpolate the intermediate reference image features. Then, tracking control laws can be designed to follow these trajectories smoothly.

5 Conclusion

In this paper, we have proposed a new path planning algorithm in the image space for the fixed-camera configuration with only one camera. To the best of our knowledge, the proposed algorithm is the first one for dealing with the

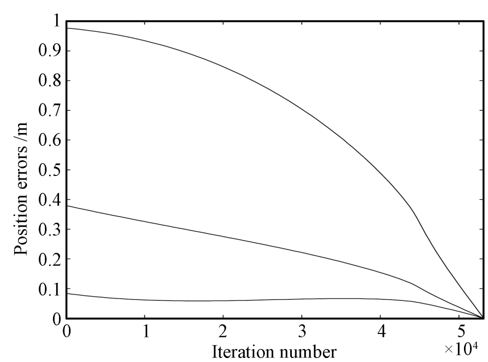


(a) Image feature error between the current and the intermediate reference image features

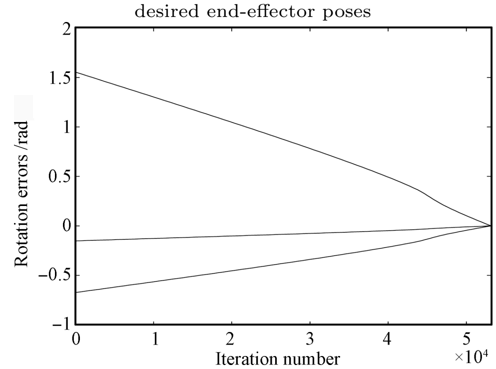


(b) Zoom in to (a)

Fig. 10 Visual servoing results for the case with 80 intermediate points



(a) Translational errors between the current and the constant desired end-effector poses



(b) Rotational errors between the current and the constant desired end-effector poses

Fig. 11 Visual servoing results for the case with 80 intermediate points

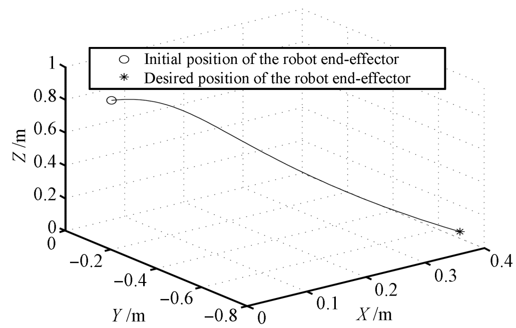


Fig. 12 Visual servoing results for the case with 80 intermediate points: position trajectory of the robot end-effector in 3D space

problem of image-space path planning with a single, uncalibrated and fixed camera, and this is the main novelty and contribution of this paper. Given the initial and desired image features of the robot end-effector, the proposed algorithm can automatically generate intermediate reference image features to connect the initial with the desired image features. To successfully accomplish the planning task, two additional sets of image features, which are captured when the end-effector is respectively located at a pure translation and a pure rotation apart from its initial position, should be collected to recover the projective motion parameters of the robot end-effector between its initial and desired poses. Note that this process can also be fulfilled automatically without any knowledge of the camera parameters and the target model.

To show the necessity and feasibility of the proposed algorithm, simulation results using a PUMA560 robot manipulator are also presented, from which we can know that the robot end-effector can be controlled to its desired destination successfully by using an image-based visual servoing approach with the help of the proposed algorithm. Conversely, without the use of the proposed algorithm, the robot end-effector fails to arrive at its desired destination. In addition, every set of generated intermediate reference image features can be attained by the actual image features of the end-effector with an arbitrary small error, which means that the planned intermediate reference image features correspond to physically valid 3D rigid motion of the robot end-effector and can be feasible. It is also found that the robot joint limits can be avoided to a certain extent by planning sufficient numbers of intermediate reference image features. It should be pointed out that the disadvantage of the proposed algorithm is that it cannot absolutely guarantee the robot joint-limits avoidance and the image features cannot definitely be kept in the camera′s field of view all the time. Hence, the objective in our future research is to completely solve these problems by taking the robot joint-limits, camera's field of view and self-occlusions constraints into consideration in the formulation of the image-space path planning algorithm.

References

1 Mezouar Y, Chaumette F. Model-free optimal trajectories in the image space. In: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Maui, Hawaii: IEEE, 2001. 25−30

2 Mezouar Y, Chaumette F. Optimal camera trajectory with image-based control. *International Journal of Robotics Research*, 2003, **22**(10−11): 781−803

3 Mezouar Y, Chaumette F. Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation*, 2002, **18**(4): 534−549

4 Mezouar Y, Chaumette F. Avoiding self-occlusions and preserving visibility by path planning in the image. *Robotics and Autonomous Systems*, 2002, **41**(2−3): 77−87

5 Mezouar Y, Remazeilles A, Gros P, Chaumette F. Images interpolation for image-based control under large displacement. In: Proceedings of the 2002 IEEE International Conference on Robotics and Automation. Washington DC, USA: IEEE, 2002. 3787−3794

6 Allotta B, Fioravanti D. 3D motion planning for image-based visual servoing tasks. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation. Barcelona, Spain: IEEE, 2005. 2173−2178

7 Chesi G, Hung Y S. Global path-planning for constrained and optimal visual servoing. *IEEE Transactions on Robotics*, 2007, **23**(5): 1050−1060

8 Chesi G, Prattichizzo D, Vicino A. Straight line path-planning in visual servoing. *Journal of Dynamic Systems, Measurement, and Control*, 2007, **129**(4): 541−543

9 Chesi G. Visual servoing path planning via homogeneous forms and LMI optimizations. *IEEE Transactions on Robotics*, 2009, **25**(2): 281−291

10 Arvani F, Mann G K I, Fisher A, Gosine R G. Sampling-based path planning for robust feature-based visual servoing. In: Proceedings of the 2009 Canadian Conference on Electrical and Computer Engineering. St. John′s: IEEE, 2009. 823−826

11 Schramm F, Morel G. A calibration free analytical solution to image points path planning that ensures visibility. In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation. New Orleans, LA: IEEE, 2004. 485−490

12 Schramm F, Morel G. Ensuring visibility in calibration-free path planning for image-based visual servoing. *IEEE Transactions on Robotics*, 2006, **22**(4): 848−854

13 Schramm F, Geffard F, Morel G, Micaelli A. Calibration free image point path planning simultaneously ensuring visibility and controlling camera path. In: Proceedings of the 2007 IEEE International Conference on Robotics and Automation. Roma, Italy: IEEE, 2007. 2074−2079

14 Schramm F, Micaelli A, Morel G. Calibration free path planning for visual servoing yielding straight line behaviour both in image and work space. In: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. Edmonton, Canada: IEEE, 2005. 2216−2221

15 Malis E. Visual servoing invariant to changes in camera-intrinsic parameters. *IEEE Transactions on Robotics and Automation*, 2004, **20**(1): 72−81

16 Park J S, Chung M J. Image space trajectory generation for image-based visual servoing under large pose error. In: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Maui, Hawaii: IEEE, 2001. 1159−1164

17 Park J S, Chung M J. Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy. *IEEE Transactions on Robotics and Automation*, 2003, **19**(2): 250−258

18 Park J S, Chung M J. Image space path planning in consideration of mechanical constraints for image-based visual servoing. In: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. Las Vegas, Nevada: IEEE, 2003. 755−760

19 Hutchinson S, Hager G D, Corke P I. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 1996, **12**(5): 651−670

20 Deng L F, Janabi-Sharifi F, Wilson W J. Hybrid motion control and planning strategies for visual servoing. *IEEE Transactions on Industrial Electronics*, 2005, **52**(4): 1024−1040

21 Hosoda K, Asada M. Versatile visual servoing without knowledge of true Jacobian. In: Proceedings of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems. Munich, Germany: IEEE, 1994. 186−193

22 Corke P I. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 1996, **3**(1): 24−32

LIANG Xin-Wu Postdoctor in the Department of Automation, Shanghai Jiao Tong University. He received his bachelor and Ph. D. degrees from the Department of Control Science and Engineering, Huazhong University of Science and Technology in 2006 and 2011, respectively. His research interest covers robot control, visual servoing, adaptive control, and computer vision. Corresponding author of this paper. E-mail: xinwu113@163.com

HUANG Xin-Han Professor in the Department of Control Science and Engineering, Huazhong University of Science and Technology. He joined the Robotics Institute of Carnegie-Mellon University at Pittsburgh, USA as a visiting scholar from 1985 to 1986 and the Systems Engineering Division of Wales University at Cardiff, UK as a senior visiting scholar in 1996. His research interest covers robotics, sensing techniques, data fusion, and intelligent control.
E-mail: xhhuang@mail.hust.edu.cn

WANG Min Received her bachelor and master degrees from the Department of Automatic Control Engineering of Huazhong University of Science and Technology (HUST) in 1982 and 1989, respectively. She is a professor in the Department of Control Science and Engineering of HUST. Her research interest covers robotics, sensing techniques, and neural networks and its applications.
E-mail: wm526@163.com