Visual Servoing Path-Planning for Cameras Obeying the Unified Model

Tiantian Shen, Graziano Chesi

Department of Electrical and Electronic Engineering, the University of Hong Kong, Hong Kong

tiantianshen@gmail.com, chesi@eee.hku.hk

Abstract

This paper proposes a path-planning visual servoing strategy for a class of cameras that includes conventional perspective cameras, fisheye cameras, and catadioptric cameras as special cases. Specifically, these cameras are modeled by adopting a unified model recently proposed in the literature, and the strategy consists of designing image trajectories for eye-in-hand robotic systems that allow the robot to reach a desired location while satisfying typical visual servoing constraints. To this end, the proposed strategy introduces the projection of the available image features onto a virtual plane and the computation of a feasible image trajectory through polynomial programming. Then, the computed image trajectory is tracked by using an image-based visual servoing controller. Experimental results with a fisheye camera mounted on a 6-DOF robot arm are presented in order to illustrate the proposed strategy.

keywords: visual servoing, path planning, unified model,

1 INTRODUCTION

Visual servoing is a control technique to guide the motion of a robot with respect to a target by exploiting visual information obtained by a camera. Various visual servoing methods have been proposed and are classified according to the error function as position-based visual servoing (PBVS) [1] and image-based visual servoing (IBVS) [2]. PBVS approaches regulate pose error and has no control in the image plane, while IBVS methods directly transfer image feature errors into camera velocities. The strengths and weaknesses of PBVS and IBVS techniques are discussed in [3]. A lot of work have been done for the combination of PBVS and IBVS techniques to avoid respective drawbacks. These methods include partition of the degrees of freedom [4], 2.5 D visual servoing [5] and hybrid switched-system [6]. Navigation functions [7] and circular-like trajectories [8] are proposed to enlarge the convergence domain. Optimal object configurations are explored to minimize the positioning error [9]. However, it is still a difficult problem to consider constraints, such as maintaining all the features in the field of view of the camera while the camera displacement is very large, keeping the robot endpoint and its joints within their physical limits, and etc. Constraints in both the image and pose domain motivate the integration of path-planning into visual servoing [10, 11]. Existing path-planning techniques for visual servoing have been divided into four categories, the benefits and drawbacks of which are discussed in a review [12]. For other advanced approaches, see the surveys [13, 14] and the collection [15].

These approaches and techniques are mainly proposed for conventional perspective cameras. On the other hand, cameras with large field of view, such as fisheye cameras (which combine perspective cameras and fisheye lens) and catadioptric cameras (which consist of perspective cameras and mirrors) are promising sensors to be used in robotic tasks. These cameras can be modeled by adopting a unified model recently proposed in the literature, see for instance [16]. A visual servoing and pose estimation method has recently been proposed for cameras obeying the unified model (which includes non-perspective and conventional perspective cameras), where a virtual servoing process is performed [17]. However, visual servoing path-planning for such cameras has not yet been proposed in the literature. This paper makes the first attempt to perform path-planning to take into account multiple constraints for non-perspective cameras.

In this paper, a visual servoing path-planning strategy for multi-constraint satisfaction is proposed for cameras obeying the unified model. In particular, the proposed strategy considers typical constraints such as keeping the object features in the field of view of the camera and keeping the robot away from collision with the obstacle. The methodology consists of the following steps. First, the available image features are reprojected onto a virtual plane independently on the type of camera used to acquire them. Second, a feasible image trajectory is computed by exploiting the obtained virtual image features through polynomial programming. Lastly, the computed image trajectory is tracked by using an IBVS controller. The proposed strategy is illustrated and validated through experimental results with a fisheye camera mounted on a 6-DOF robot arm. A preliminary version of this paper appeared in [18].

The paper is organized as follows. Section 2 introduces the unified camera model and the problem formulation. Section 3 describes the proposed strategy for visual servoing path-planning with cameras obeying the unified model. Section 4 reports the experimental results. Lastly, Section 5 concludes the paper with some final remarks.

2 PRELIMINARIES

Let us introduce the following notation. $\mathscr{R}$ is the real number set, $\mathbf{I}_n$ is the $n \times n$ identity matrix, $\mathbf{e}_i$ is the i-th column of $\mathbf{I}_3$, $\mathbf{0}_n$ is the $n \times 1$ null vector, $\mathbf{1}_n$ is the $n \times 1$ vector with all elements equal to 1, $SO(3)$ is the set of 3×3 rotation matrices, $[\mathbf{t}]_\times$ is the skew-symmetric matrix of vector $\mathbf{t} \in \mathscr{R}^3$, $\|\mathbf{w}\|$ is the Euclidean norm of vector $\mathbf{w}$.

Let $\mathscr{F}^\circ = \{\mathbf{R}^\circ, \mathbf{t}^\circ\}$, $\mathscr{F}^* = \{\mathbf{R}^*, \mathbf{t}^*\}$ be two camera frames in the 3D space, their relative camera pose $\{\mathbf{R}, \mathbf{t}\}$ expressed in the first camera frame $\mathscr{F}^\circ$ is:

$$\begin{cases} \mathbf{R} = \mathbf{R}^{\circ\top} \mathbf{R}^* \\ \mathbf{t} = \mathbf{R}^{\circ\top} \left(\mathbf{t}^* - \mathbf{t}^\circ \right), \end{cases} \tag{1}$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathscr{R}^3$ define respectively the relative orientation and translation. In the sequel, we call $\mathscr{F}^\circ$ the initial camera frame and let it be the reference frame $\{\mathbf{I_3}, \mathbf{0_3}\}$, while $\mathscr{F}^*$ the desired one $\{\mathbf{R}, \mathbf{t}\}$.

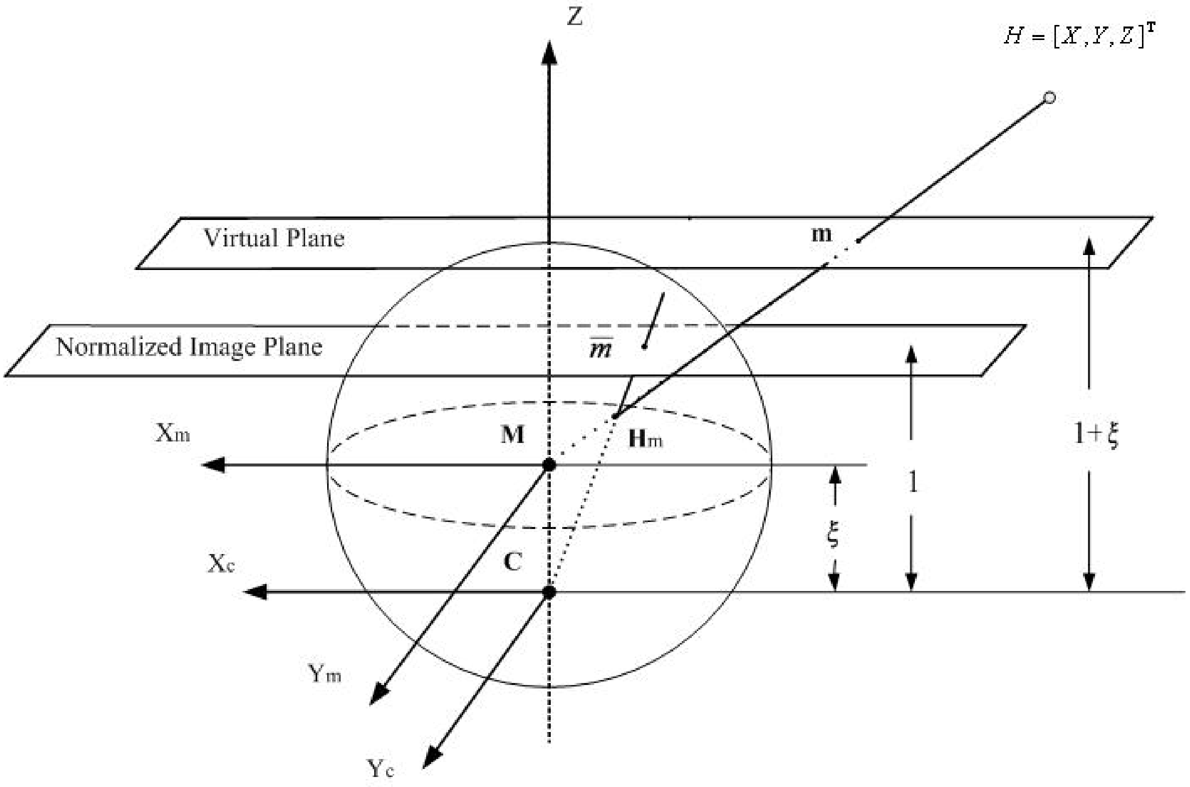


Figure 1: The unified model and the virtual plane.

In cameras obeying the unified model, image projection of a target point $\mathbf{H}$ is formulated as $\bar{\mathbf{p}} = \mathbf{K}\bar{\mathbf{m}}$, where $\mathbf{K}$ is the camera intrinsic parameters matrix, and

$$\bar{\mathbf{m}} = \left(\frac{X}{Z + \xi \|\mathbf{H}\|}, \frac{Y}{Z + \xi \|\mathbf{H}\|}, 1 \right)^{\top}. \tag{2}$$

To fully explain it, see the unified model in Fig. 1. There is an unitary sphere centered at $\mathbf{M}$. Coordinates $[X,Y,Z]^{\top}$ of point $\mathbf{H}$ is expressed with respect to the sphere center, coefficient ξ denotes the distance between optical center $\mathbf{C}$ and sphere center along the z-axis. Expression (2) is built by two projections: the first one is projection of $\mathbf{H}$ on the unitary sphere at $\mathbf{H}_m$, followed by a second projection from optical center on the normalized image plane at $\bar{\mathbf{m}}$.

Problem. The problem considered in this paper is how to motivate a robot mounted with camera obeying the unified model from the initial pose to the desired one while satisfying multiple constraints.

3 PROPOSED STRATEGY

A path-planning strategy is proposed for robot with non-perspective cameras to satisfy multiple constraints such as collision avoidance and visibility preservation during the servo. To this end, we introduce a virtual plane and conduct path-planning on this virtual plane. Afterwards, the planned path is followed by an image-based visual servoing controller.

3.1 Introduction of Virtual Plane

As shown in Fig. 1, the virtual plane is located tangent with the unitary sphere, therefore it has a displacement of 1 from sphere center along the z-axis. Projection on the virtual plane, which we denote as $\mathbf{m}$, may be developed from $\bar{\mathbf{m}}$ as follows:

$$\bar{\mathbf{m}} = (\bar{m}_1, \bar{m}_2, 1)^\top, \tag{3}$$

$$\mathbf{m} = (\gamma\bar{m}_1, \gamma\bar{m}_2, 1)^\top, \tag{4}$$

$$\gamma = \frac{1+\xi\delta}{1-\xi^2(\bar{m}_1^2+\bar{m}_2^2)}, \tag{5}$$

$$\delta = \sqrt{1+(1-\xi^2)(\bar{m}_1^2+\bar{m}_2^2)}. \tag{6}$$

When the model coefficient ξ draws to zero, the normalized image plane and the virtual plane overlap, $\bar{\mathbf{m}}$ and $\mathbf{m}$ coincide with each other, and the unified model boils down to the perspective camera model.

Suppose a set of 3D points are projected onto a camera obeying the unified model, and $\mathbf{h}_j, j = 1, \dots n$ are their coordinates in the initial reference camera frame. In that case, their projections in the initial and desired cameras are respectively expressed as follows:

$$\bar{\mathbf{p}}_j^\circ = \left(\frac{\mathbf{e}_1^\top \mathbf{K}\mathbf{h}_j}{\mathbf{e}_3^\top \mathbf{h}_j + \xi \|\mathbf{h}_j\|}, \frac{\mathbf{e}_2^\top \mathbf{K}\mathbf{h}_j}{\mathbf{e}_3^\top \mathbf{h}_j + \xi \|\mathbf{h}_j\|}, 1 \right)^\top, \tag{7}$$

$$\bar{\mathbf{p}}_j^* = \left(\frac{\mathbf{e}_1^\top \mathbf{K}\mathbf{R}^\top(\mathbf{h}_j - \mathbf{t})}{\mathbf{e}_3^\top \mathbf{R}^\top(\mathbf{h}_j - \mathbf{t}) + \xi \|\mathbf{h}_j - \mathbf{t}\|}, \frac{\mathbf{e}_2^\top \mathbf{K}\mathbf{R}^\top(\mathbf{h}_j - \mathbf{t})}{\mathbf{e}_3^\top \mathbf{R}^\top(\mathbf{h}_j - \mathbf{t}) + \xi \|\mathbf{h}_j - \mathbf{t}\|}, 1 \right)^\top. \tag{8}$$

Their values on the normalized image plane are:

$$\bar{\mathbf{m}}_j^\circ = \mathbf{K}^{-1}\bar{\mathbf{p}}_j^\circ, \tag{9}$$

$$\bar{\mathbf{m}}_j^* = \mathbf{K}^{-1}\bar{\mathbf{p}}_j^*. \tag{10}$$

Through development (3)∼(6), their projections on the virtual plane: $\mathbf{m}_j^\circ$ and $\mathbf{m}_j^*$ are obtained. Accordingly, virtual image projections:

$$\mathbf{p}_j^\circ = \mathbf{K}\mathbf{m}_j^\circ, \tag{11}$$

$$\mathbf{p}_j^* = \mathbf{K}\mathbf{m}_j^* \tag{12}$$

are computed for subsequent motion and structure estimation.

3.2 Motion and Structure Estimation

Virtual image projections of the target in the initial and desired cameras are used to estimate the camera motion and target structure. Eight-point algorithm by exploiting the essential matrix [19] is first applied to estimate the camera pose, in the case of which we need at least eight non-planar points as the target. After that, target structure is estimated via minimization of algebraic error. In order to improve the estimation results, further minimization of 2D reprojection error on the virtual plane is performed.

A possible way for error minimization is to solve one while fixing the others among three variables: camera rotation, translation and target structure. These variables are updated alternatively to approach a possible minimum point of the 2D reprojection error.

In this paper, we propose to integrate camera rotation, translation and 3D coordinates of all the target points into one vector variable:

$$\mathbf{s} = \left(\mathbf{s}_{\mathbf{R}}^{\top}, \mathbf{s}_{\mathbf{t}}^{\top}, \mathbf{s}_{\mathbf{h}_1}^{\top}, \mathbf{s}_{\mathbf{h}_2}^{\top}, \dots, \mathbf{s}_{\mathbf{h}_n}^{\top}\right)^{\top}, \tag{13}$$

where $\mathbf{s}_{\mathbf{R}}$ is vector representation of the rotation matrix.

Initial value of variable $\mathbf{s}$ comes from the solution of essential matrix algorithm and minimization of algebraic error: camera motion $\{\mathbf{R}, \mathbf{t}\}$ and target coordinates $\mathbf{h}_j, j = 1, \dots n$:

$$\mathbf{s}_{\mathbf{R}} = \mathbf{q}_1(\mathbf{R}) \in \mathscr{R}^4 \quad \text{or} \quad \mathbf{q}_2(\mathbf{R}) \in \mathscr{R}^3, \tag{14}$$

$$\mathbf{s}_{\mathbf{t}} = \mathbf{t}, \tag{15}$$

$$\mathbf{s}_{\mathbf{h}_j} = \mathbf{h}_j, j = 1, \dots, n. \tag{16}$$

In the above equations, $\mathbf{q}_1(\mathbf{R})$, $\mathbf{q}_2(\mathbf{R})$ are respectively quaternion and Cayley [20] representation of the rotation matrix:

$$\mathbf{q}_1(\mathbf{R}) = \left(\sin\frac{\theta}{2}\mathbf{v}^{\top}, \cos\frac{\theta}{2}\right)^{\top}, \tag{17}$$

$$\mathbf{q}_2(\mathbf{R}) = \left[(\mathbf{R} - \mathbf{I}_3) \cdot (\mathbf{R} + \mathbf{I}_3)^{-1}\right]_{\otimes}, \tag{18}$$

where $\theta \in (0, \pi)$ and $\mathbf{v} \in \mathscr{R}^3 : \|\mathbf{v}\| = 1$ are rotation angle and axis such that $\mathbf{R} = e^{\theta[\mathbf{v}]_{\times}}$, $[\mathbf{M}]_{\otimes}$ is a 3×1 vector $(M_{32}, M_{13}, M_{21})^{\top}$ generated from the 3×3 skew symmetric matrix $\mathbf{M}$, the entry of which at the i-th row and the j-th column is symbolized as M_{ij}. Despite of the degenerating case when $\theta = \pi$ in Cayley representation, it is tolerable since the target is always required to be kept in front of the camera in teaching-by-showing visual servoing.

Starting from the initial value of variable $\mathbf{s}$, it is refreshed through multidimensional unconstrained nonlinear minimization of the 2D reprojection error:

$$\min_{\mathbf{s}} \sum_{j=1}^{n} \left(\left\| \mathbf{p}_j^{\circ} - \mathbf{K}\frac{\mathbf{h}_j}{\mathbf{e}_3^{\top}\mathbf{h}_j} \right\|^2 + \left\| \mathbf{p}_j^{*} - \frac{\mathbf{K}\mathbf{R}^{\top}(\mathbf{h}_j - \mathbf{t})}{\mathbf{e}_3^{\top}\mathbf{R}^{\top}(\mathbf{h}_j - \mathbf{t})} \right\|^2 \right). \tag{19}$$

With newly obtained variable $\mathbf{s}$, camera rotation is retrieved as follows:

$$\begin{aligned}
\mathbf{q}_1(\mathbf{R}) &= \frac{\mathbf{s}_{\mathbf{R}}}{\|\mathbf{s}_{\mathbf{R}}\|} = (a_1, a_2, a_3, a_4)^{\top}, \\
\mathbf{R} &= \Lambda(\mathbf{q}_1(\mathbf{R})) \\
&= \begin{pmatrix} a_1^2 - a_2^2 - a_3^2 + a_4^2 & 2(a_1a_2 - a_3a_4) & 2(a_1a_3 + a_2a_4) \\ 2(a_1a_2 + a_3a_4) & -a_1^2 + a_2^2 - a_3^2 + a_4^2 & 2(a_2a_3 - a_1a_4) \\ 2(a_1a_3 - a_2a_4) & 2(a_2a_3 + a_1a_4) & -a_1^2 - a_2^2 + a_3^2 + a_4^2 \end{pmatrix},
\end{aligned} \tag{20}$$

or

$$\begin{aligned} &\mathbf{q}_2(\mathbf{R}) = \mathbf{s}_\mathbf{R} \in \mathscr{R}^3, \\ &\mathbf{R} = \left(\mathbf{I}_3 - [\mathbf{q}_2(\mathbf{R})]_\times\right)^{-1} \cdot \left(\mathbf{I}_3 + [\mathbf{q}_2(\mathbf{R})]_\times\right). \end{aligned} \tag{21}$$

No matter what kind of rotation representation is applied, camera translation and 3D coordinates of the target may simply be restored:

$$\begin{aligned} &\mathbf{t} = \frac{\mathbf{s}_\mathbf{t}}{||\mathbf{s}_\mathbf{t}||}, \\ &\mathbf{h}_j = \mathbf{s}_{\mathbf{h}_j}, j = 1, \ldots, n. \end{aligned} \tag{22}$$

It is obvious that the ultimate results of motion and structure estimation shall satisfy image boundary condition expressed in (7) and (8).

3.3 Path-Planning Based Visual Servoing

The objective of this section is to plan the 3D path and follow it by reducing error between real trajectories and planned trajectories. In this paper, the error is computed on the virtual plane for cameras obeying the unified model. For the sake of path-planning, we introduce parameter $w \in [0,1]$ with its boundary describing the starting point and the destination. Therefore, each stage of camera motion could be related with the parameter w, that is $\{\mathbf{R}(w), \mathbf{t}(w)\}$. We propose to use groups of polynomials in w to stand for items in camera translation and rotation quaternion:

$$\begin{cases} \phi(w) = \tilde{U} \cdot [w^\sigma, w^{\sigma-1}, ..., w, 1]^\top \\ \mathbf{t}(w) = \tilde{V} \cdot [w^\tau, w^{\tau-1}, ..., w, 1]^\top, \end{cases} \tag{23}$$

where rotation quaternion $\phi(w) \in \mathscr{R}^4$ is expressed by four polynomials of w with degree σ, translation $\mathbf{t}(w) \in \mathscr{R}^3$ is decomposed into three polynomials of w with degree τ, $\tilde{U} \in \mathscr{R}^{4\times(\sigma+1)}$ and $\tilde{V} \in \mathscr{R}^{3\times(\tau+1)}$ stand for the coefficient matrices.

Some entries in the coefficient matrices $\tilde{U}$ and $\tilde{V}$ are determined by the initial camera pose $\{\mathbf{I}_3, \mathbf{0}_3\}$ and the results of motion estimation $\{\mathbf{R}, \mathbf{t}\}$ in section 3.2:

$$\begin{cases} \{\phi(0), \mathbf{t}(0)\} = \{[\mathbf{0}_3, 1]^\top, \mathbf{0}_3\} \\ \{\phi(1), \mathbf{t}(1)\} = \{\mathbf{q}_1(\mathbf{R}), \mathbf{t}\}, \end{cases} \tag{24}$$

where $\mathbf{q}_1(\mathbf{R})$ is declared in (20). Consequently, coefficient matrices satisfying pose boundary condition (24) is expressed as follows:

$$\begin{cases} \tilde{U} = [\mathbf{q}_1(\mathbf{R}) - U \cdot \mathbf{1}_{\sigma-1} - [\mathbf{0}_3^\top, 1]^\top, U, [\mathbf{0}_3^\top, 1]^\top], \\ \tilde{V} = [\mathbf{t} - V \cdot \mathbf{1}_{\tau-1}, V, \mathbf{0}_3]. \end{cases} \tag{25}$$

We denote the unknowns in coefficient matrices $\tilde{U}$ and $\tilde{V}$ as $U \in \mathscr{R}^{4\times(\sigma-1)}$ and $V \in \mathscr{R}^{3\times(\tau-1)}$ respectively. Values in U and V regulate the planned 3D camera path.

3.3.1 Multiple Constraints

Constraints on camera path are required to avoid collision, occlusion and preserve visibility. It is worth mentioning that for IBVS existing methods have been proposed to exploit the redundant degree of freedom (DOF) of a robot to meet some of the constraints [21, 22]. In this paper, however, we use path-planning method to consider multi-constraint satisfaction.

For collision avoidance, we may simply limit the workspace by setting up the following inequality:

$$\|\mathbf{t}(w) - \mathbf{obs}\| \geq d, \quad \forall w \in (0,1), \tag{26}$$

where constant d is a given safe distance between the camera trajectory $\mathbf{t}(w)$ and the obstacle, vector $\mathbf{obs} \in \mathscr{R}^3$ represents obstacle position in the reference frame. Bring (23) into (26), we have

$$\left(w^{\tau}, w^{\tau-1}, ..., w, 1\right) \tilde{\mathscr{V}}^{\top} \tilde{\mathscr{V}} \left(w^{\tau}, w^{\tau-1}, ..., w, 1\right)^{\top} - d^2 \geq 0, \tag{27}$$

where

$$\tilde{\mathscr{V}} = [\mathbf{t} - V \cdot \mathbf{1}_{\tau-1}, V, -\mathbf{obs}]. \tag{28}$$

The planned camera path shall keep an appropriate large distance from the point representing the obstacle when inequality (27) is satisfied. Two or more points may also be used to stand for an obstacle and form a group of polynomial inequalities.

Although the obstacle avoidance is accomplished in (27), it would probably cause the camera to lose sight of the target and fail the servo task. Therefore, visibility constraint is incorporated to ensure image projections of the target are always available. We express coordinates of image projections as $\bar{\mathbf{p}}_j = [\bar{x}_j, \bar{y}_j, 1]^{\top}$, then the visibility requirement along the planned path may be fulfilled by:

$$\begin{cases} Z_j(w) > 0, \forall w \in (0,1), j = 1, ..., n, & (29) \\ 0 \leq \bar{x}_j(w) \leq \bar{\zeta}_x, & (29') \\ 0 \leq \bar{y}_j(w) \leq \bar{\zeta}_y, & (29'') \\ (\bar{x}_j(w) - \bar{x}_o)^2 + (\bar{y}_j(w) - \bar{y}_o)^2 \geq \bar{\zeta}_o^2, & (29''') \end{cases}$$

where Z_j is the depth of the j-th point with respect to camera frame, $\bar{\zeta}_x$ and $\bar{\zeta}_y$ denote the image size along the x and y coordinate respectively, $\bar{x}_o$ and $\bar{y}_o$ denote the image center, $\bar{\zeta}_o$ signifies the radius of dead zone at the image center formed in catadioptric imaging systems.

For cameras obeying the unified model, constraints (29′)-(29‴) may be transferred onto the virtual plane introduced in section 3.1. We symbolize coordinates of the corresponding projection of $\bar{\mathbf{p}}_j$ on the virtual plane as $\mathbf{m}_j = [m_{j1}, m_{j2}, 1]^{\top}$, and its boundary ζ_x, ζ_y according to the image size $\bar{\zeta}_x, \bar{\zeta}_y$ could be established:

$$\begin{pmatrix} \zeta_x \\ \zeta_y \\ 1 \end{pmatrix} = \begin{pmatrix} \gamma & 0 & 0 \\ 0 & \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{K}^{-1} \begin{pmatrix} \bar{\zeta}_x \\ \bar{\zeta}_y \\ 1 \end{pmatrix}, \tag{30}$$

where γ is defined in (5). In this case, constraints (29′) and (29″) may be transformed into:

$$\begin{cases} -\zeta_x < m_{j1}(w) < \zeta_x, \\ -\zeta_y < m_{j2}(w) < \zeta_y. \end{cases} \tag{31}$$

Similar transformation is applied to (29‴):

$$m_{j1}(w)^2 + m_{j2}(w)^2 \geq \zeta_o^2, \tag{32}$$

where ζ_o is the corresponding value of radius $\bar{\zeta}_o$ on the virtual plane. As a result, visibility constraint may be met by a series of polynomial inequalities:

$$\begin{cases} \mathbf{e}_3^\top \mathbf{c} > 0, & (33) \\ \zeta_x^2(\mathbf{e}_3^\top \mathbf{c})^2 - (\mathbf{e}_1^\top \mathbf{c})^2 > 0, & (33') \\ \zeta_y^2(\mathbf{e}_3^\top \mathbf{c})^2 - (\mathbf{e}_2^\top \mathbf{c})^2 > 0, & (33'') \\ (\mathbf{e}_1^\top \mathbf{c})^2 + (\mathbf{e}_2^\top \mathbf{c})^2 - \zeta_o^2(\mathbf{e}_3^\top \mathbf{c})^2 \geq 0, & (33''') \end{cases}$$

where

$$\mathbf{c} = \Lambda(\phi(w))^\top(\mathbf{h}_j - \mathbf{t}(w)), \quad j = 1, ..., n. \tag{34}$$

Polynomial degrees are $2\sigma + \tau$ in (33) and $4\sigma + 2\tau$ in (33′)-(33‴) according to the representation in (23). If the equipped sensor is not a catadioptric camera, constant ζ_o shall be assigned to be zero.

We put together all the constraints in (27) and (33)-(33‴), and describe them with an integrated function:

$$\begin{aligned} & g_{k,w,j}(U,V) > 0, \\ & \forall k = 1, ..., k_{MAX}, \forall w \in (0,1), \forall j = 1, ..., n. \end{aligned} \tag{35}$$

It is straightforward to interpret that assembling function $g(U,V)$ shall be positive for all $k = 1, ..., k_{MAX}, w \in (0,1)$ and $j = 1, ..., n$.

3.3.2 Computing and Following the Path

Appropriate values of the unknowns U and V shall be found in (35) for the planned path, and the optimization method is presented immediately.

We integrate all the entries of U and V matrices into a united variable $\mathbf{v} \in \mathscr{R}^{4(\sigma-1)+3(\tau-1)}$ and assign $\mathbf{0}_{4(\sigma-1)+3(\tau-1)}$ to be its initial value, which indicates a straight line as the camera 3D path. With initial value in $\mathbf{v}$, the minimum value of the left hand side of all the constraints in (35) are calculated:

$$\begin{aligned} g_{1,w,j}(\mathbf{v}) &= \min_{w\in(0,1)} \left[\|\mathbf{t}(w) - \mathbf{obs}\|^2 - d^2 \right], \\ g_{2,w,j}(\mathbf{v}) &= \min_{w\in(0,1)} \min_{j=1,...,n} \mathbf{e}_3^\top \mathbf{c}, \\ g_{3,w,j}(\mathbf{v}) &= \min_{w\in(0,1)} \min_{j=1,...,n} \left[\zeta_x^2(\mathbf{e}_3^\top \mathbf{c})^2 - (\mathbf{e}_1^\top \mathbf{c})^2 \right], \\ g_{4,w,j}(\mathbf{v}) &= \min_{w\in(0,1)} \min_{j=1,...,n} \left[\zeta_y^2(\mathbf{e}_3^\top \mathbf{c})^2 - (\mathbf{e}_2^\top \mathbf{c})^2 \right], \\ g_{5,w,j}(\mathbf{v}) &= \min_{w\in(0,1)} \min_{j=1,...,n} \left[(\mathbf{e}_1^\top \mathbf{c})^2 + (\mathbf{e}_2^\top \mathbf{c})^2 - \zeta_o^2(\mathbf{e}_3^\top \mathbf{c})^2 \right]. \end{aligned} \tag{36}$$

Again, a minimum value among (36) is found and defined as:

$$G(\mathbf{v}) = \min_{k=1,...,k_{MAX}} \{g_{k,w,j}(\mathbf{v})\}. \tag{37}$$

If $G(\mathbf{v})$ is positive, then all the constraints are already satisfied with initial value of $\mathbf{v}$, otherwise maximization of $G(\mathbf{v})$ is performed until it is positive.

$$G^* = \max_{\mathbf{v}} G(\mathbf{v}). \tag{38}$$

There would be more than one camera path satisfy multiple constraints collected in (35). One can simply adopt the solution returned by the solver. Since $\mathbf{v}$ is an integrated variable, local maximum of $G(\mathbf{v})$ is guaranteed. We may recover U and V matrices from the newly obtained variable.

The planned image trajectory can thus be found by the following computation:

$$\bar{\mathbf{p}}_j(w) = \frac{\mathbf{K}\Lambda(\phi(w))^\top(\mathbf{h}_j - \mathbf{t}(w))}{\mathbf{e}_3^\top \Lambda(\phi(w))^\top(\mathbf{h}_j - \mathbf{t}(w)) + \xi \|\mathbf{h}_j - \mathbf{t}(w)\|}. \tag{39}$$

Their projections on the virtual plane $\mathbf{m}_j(w)$ may be developed according to (3)-(6), or directly calculated as:

$$\mathbf{m}_j(w) = \frac{\Lambda(\phi(w))^\top(\mathbf{h}_j - \mathbf{t}(w))}{\mathbf{e}_3^\top \Lambda(\phi(w))^\top(\mathbf{h}_j - \mathbf{t}(w))}. \tag{40}$$

To track the planned path, we select

$$\mathbf{p}_j(w) = \mathbf{K}\mathbf{m}_j(w) \tag{41}$$

as features to be chased and let trajectory parameter w be the function of time:

$$w = 1 - e^{-t\lambda_1}. \tag{42}$$

At the beginning of visual servoing, $w = 0$; as time goes, w gets closer to the value of 1. With this new definition of trajectory parameter, the camera may chase features $\mathbf{p}_j(w)$ with the following instant velocity:

$$T_c = -\lambda_2 \hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}_p(w)) + \lambda_1 e^{-t\lambda_1} \hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}_p(w)}{\partial w}, \tag{43}$$

where λ_1 and λ_2 are positive gains, $\hat{\mathbf{L}}^+$ is the pseudo-inverse of the estimated image Jacobin [10], $\mathbf{s}(t)$ is the current features according to the current robot/camera position and $\mathbf{s}_p(w)$ is the planned features at time t. Both $\mathbf{s}(t)$ and $\mathbf{s}_p(w)$ are projections on the virtual plane.

The depth of each feature point used in the image Jacobin are calculated as:

$$\mathfrak{z}_j = \alpha \mathbf{e}_3^\top \mathbf{R}_c(\mathbf{h}_j - \mathbf{t}_c), \quad j = 1, \ldots, n, \tag{44}$$

where α is an estimated constant symbolizing the distance between the initial and the desired camera poses, $\mathbf{h}_j$ is 3D coordinates of the j-th point obtained in Section 3.2, $\mathbf{R}_c$ and $\mathbf{t}_c$ are the current camera pose with respect to the initial camera frame, which is estimated from the initial and the current views.

4 EXPERIMENTAL EXAMPLES

The robot used for the experiments is a Staubli RX60 6-DOF articulated arm, while the fisheye camera mounted on the robot end-effector consists of a video camera Prosilica CV640 and a fisheye lens Fujinon FE185C046HA-1. The Omnidirectional Calibration Toolbox [23] was used to calibrate the fisheye camera with the estimated intrinsic parameters and coefficient ξ in the unified model as follows:

$$\begin{aligned}\mathbf{K}_{exp} &= \begin{pmatrix} 296.9365 & 0 & 316.8076 \\ 0 & 298.4197 & 254.9012 \\ 0 & 0 & 1 \end{pmatrix}, \\ \xi_{exp} &= 1.2910.\end{aligned} \tag{45}$$

We now consider a non-planar target beside the wall with a set of 9 black dots. Example 1 and Example 2 demonstrates the proposed strategy when respectively pure translation, both translation and rotation are considered. In these examples, total computation time for motion and structure estimation and path-planning is around 6 seconds. The computation is performed on a standard PC (Pentium D CPU 3.40GHz 3.39GHz, 2.00GB of RAM, WIN XP) with MATLAB. Time consumed respectively for the two stages (motion and structure, path-planning) are given in each example.

4.1 Example 1

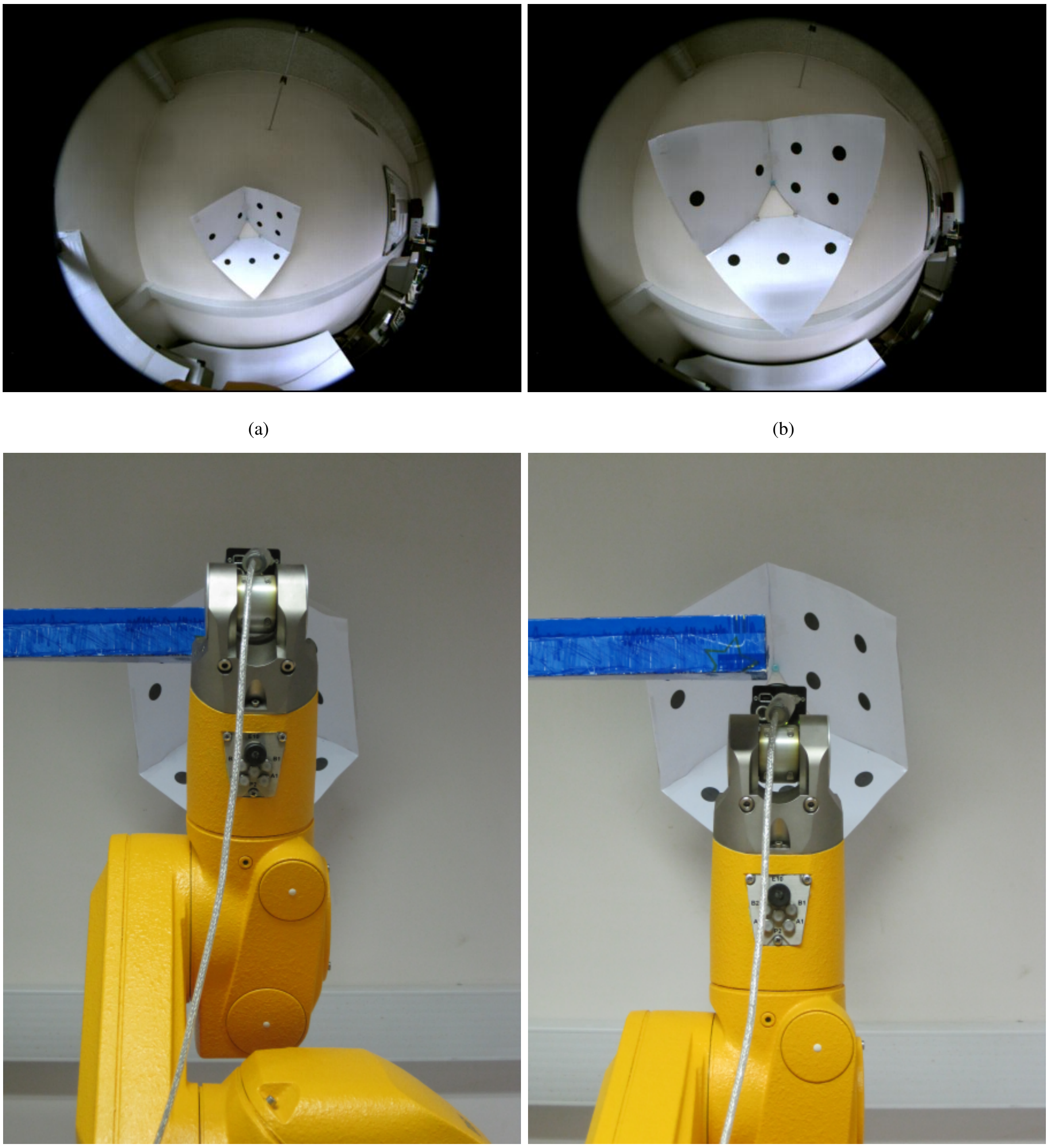
Displacement between the initial and the desired robot end-effectors is pure translation:

$$t_x = 0.0497\text{mm}, \quad t_y = -99.9782\text{mm}, \quad t_z = 199.9552\text{mm},$$

$$(u\theta)_x = -0.0030\text{deg}, \quad (u\theta)_y = 0.0013\text{deg}, \quad (u\theta)_z = -0.0015\text{deg}.$$

The images corresponding to the initial and desired poses are given in Fig. (2) (a) - (b), with image size of 659 × 432 pixels. Fig. (2) (c) - (d) show the obstacle position represented by a blue box with respect to the initial and desired camera/robot poses respectively, in this case, classical image-based visual servoing may cause collision. In order to reach the desired pose while avoiding the obstacle and keep camera view of the target at the same time, we plan the 3D path between estimated camera poses by polynomial parametrization of translation and rotation. The polynomial degrees are two for items in both translation and rotation quaternion. Visual servoing is afterwards performed to follow the planned path.

It takes about 3.073 seconds to obtain motion and structure, and then around 2.796 seconds to complete path-planning. In Fig. (3) (a), the box, as an obstacle, is quite long and limit the workspace very much. In this case, the obstacle is actually located behind and below the initial camera frame and can not be detected by the initial camera. In this particular example, the obstacle is represented by two points, the position of which are respectively approximated as $[-10,-65,85]^\top$mm and $[-5,-80,90]^\top$mm with respect to the initial camera frame of $\mathscr{F}^o$ in Fig. (3) (a). Two points with minus x-coordinates may prohibit the robot moving along the opposite direction of the x-axis. Minus y-coordinates of these two points mean the obstacle is below camera frame $\mathscr{F}^o$. Values in the z-coordinates are approximated larger than the actual position of the box since at initial pose the size of the robot arm is very large along the z-coordinate. If z-coordinates of these two obstacle points are approximated according to the actual position of the box, then the camera may avoid collision with the box, while the arm body may not. Safe distance d is selected as 55mm since near to the box the robot arm is around 100mm in the diameter. After optimization, the minimum distance between the planned path and the obstacle points is 66.127mm.



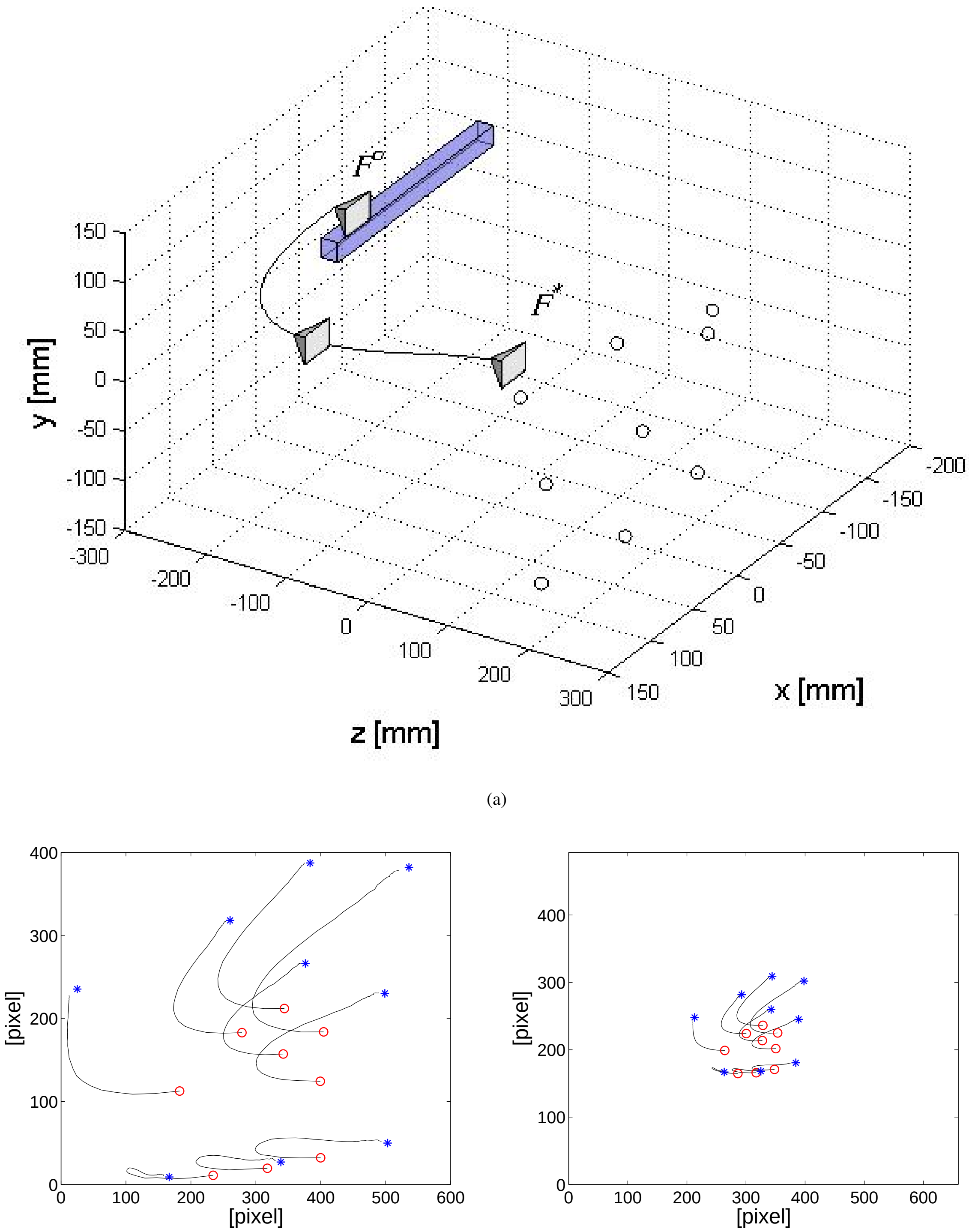
(a)

(b)

(c)

(d)

Figure 2: Experimental example 1. (a) Initial view. (b) Desired view. (c) Initial robot position. (d) Desired robot position.

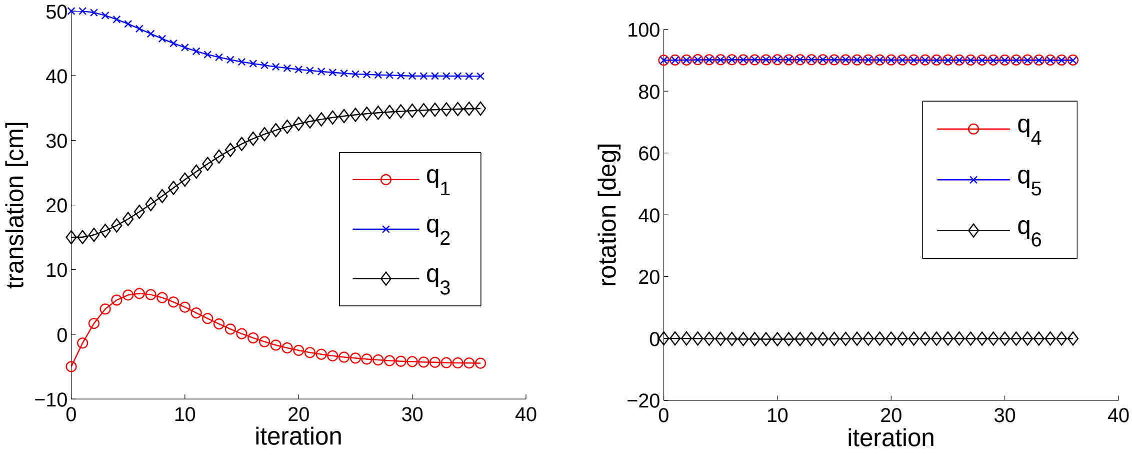


(a)

(b)

(c)

Figure 3: Experimental example 1. (a) 3D path. (b) Trajectories on the virtual plane. (c) Image trajectories.



(a)

(b)

Figure 4: Experimental example 1. (a) Robot translation evolution. (b) Robot rotation evolution.

Although the size of the robot arm is quite large along the z-coordinate, it still reaches the desired pose while avoiding the box exhibited in Fig. (3) (a). The camera path in Fig. (3) (a) is generated during the real servo, nine points plotted in front of these camera poses are estimated target structure. While performing visual servoing, trajectories on the virtual plane are obtained, as shown in Fig. (3) (b). The corresponding image trajectories are displayed in Fig. (3) (c). In Fig. (3) (b)-(c), circles in red color denote the initial projections, while blue stars the desired ones. Fig. (4) (a) - (b) plots translation and rotation evolution of the robot. At the beginning of visual servo, translation q_1 along the x-axis increases very fast to avoid collision with the box, while very slowly translation q_2 along the y-axis decreases and translation q_3 along the z-axis grows. After being away from the obstacle, translation q_1 gradually falls back to its initial value at -50 mm, q_2 decreases to the desired value of 400 mm, and q_3 keeps growing and ultimately converges to about 350 mm. Rotation angles q_4, q_5, q_6 around the x-axis, y-axis and z-axis respectively preserve constant values during the whole process of visual servoing. Results show that the robot successfully kept away from the obstacle and reached the desired pose with rotation almost unchanged. All the image trajectories generated are kept in the camera field of view, and are convergent to the desired projections.

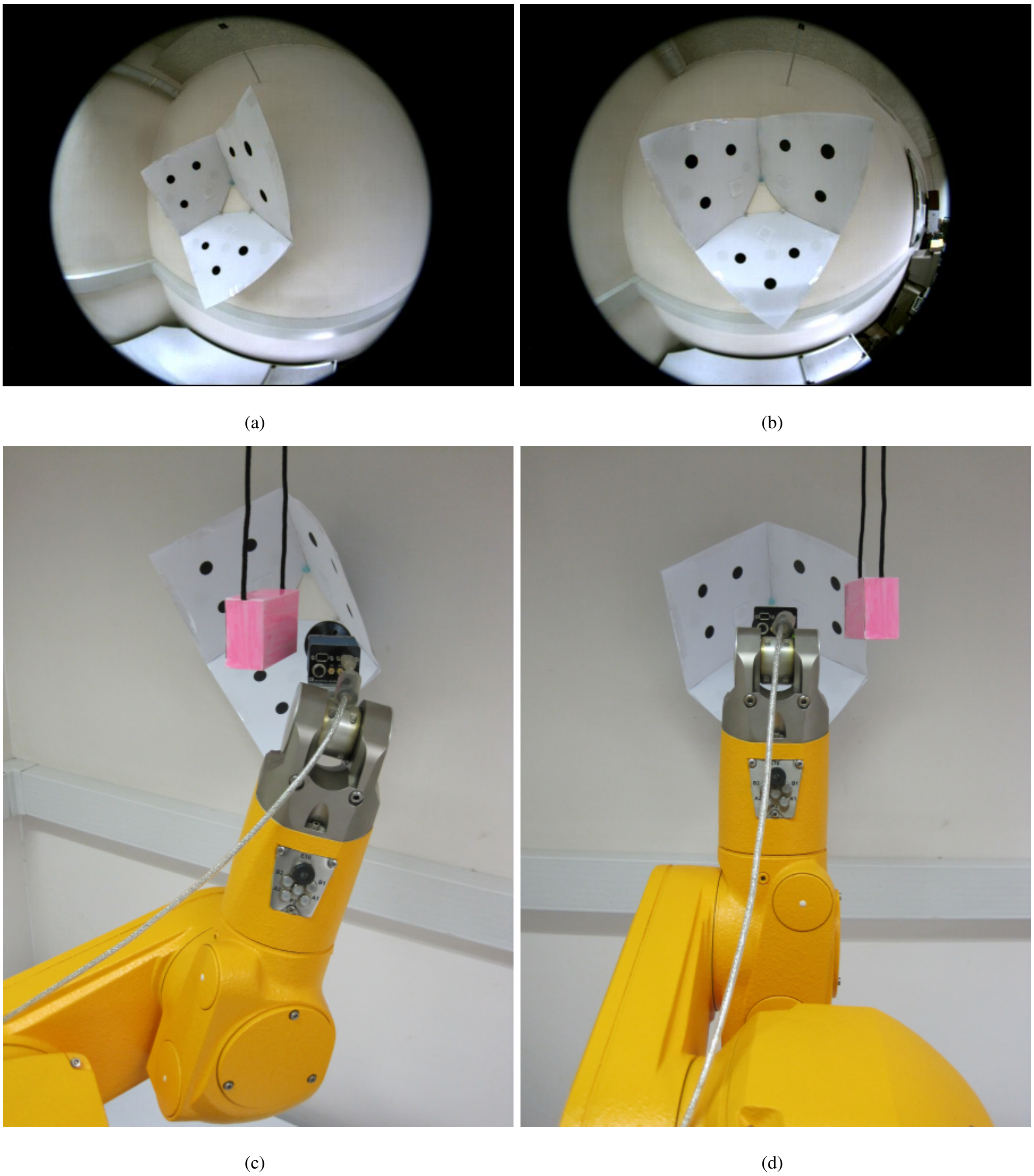
4.2 Example 2

The images corresponding to the initial and desired robot frames are given in Fig. (5) (a) - (b) respectively. Displacement between the initial and the desired robot frames:

$$t_x = -202.0935\text{mm}, \quad t_y = 0.0154\text{mm}, \quad t_z = -8.2329\text{mm},$$

$$(u\theta)_x = -0.0111\text{deg}, \quad (u\theta)_y = -24.9738\text{deg}, \quad (u\theta)_z = 0.0056\text{deg},$$

can be recognized in the snapshots shown in Fig. (5) (c) - (d). A pink box, standing for the obstacle, is very close to the initial robot/camera frame as we can see in Fig. (5) (c), which makes successful visual servoing without collision and collusion a difficult problem. Again, classical image-based visual servoing would ignore



(a)

(b)

(c)

(d)

Figure 5: Experimental example 2. (a) Initial view. (b) Desired view. (c) Initial robot position. (d) Desired robot position.

the obstacle block in the way and results in collision. In this case, an appropriate planned 3D path is required to be followed. Polynomial parametrization of translation and rotation with degrees of two is implemented to find the appropriate 3D path. Subsequently, visual servoing is applied on the virtual plane to follow the planned path.

In optimization, the obstacle is represented by one point and is approximated as $[-98,-2,40]^\top$mm expressed in the camera frame of $\mathscr{F}^o$ in Fig. (6) (a). The safe distance is selected to be 70mm since the box hanged in the scene is about 90mm long and the dimension of the fisheye camera is around 50mm in the diameter. After optimization for multi-constraints satisfaction, the planned path has the minimum distance of 82.249mm from the obstacle point. Computational time is near 3.899 seconds for motion and structure and around 1.814 seconds for path-planning. Fig. (6) (a) demonstrates camera path in 3D space avoiding collision and occlusion. The generated trajectories on virtual plane and fisheye image plane are plotted in Fig. (6) (b) - (c) respectively. It is obvious that all the image trajectories are within the image boundary, although their projections spread greatly on the virtual plane. The convergence of robot translation and rotation can be seen in Fig. (7) (a) - (b). Translation q_1 along the x-axis keeps falling down to the desired value of -50 mm. The shape of evolution q_2 is attributed to the requirement of obstacle avoidance. Ultimate values in robot translation along the y-axis and z-axis are almost the same with their starting values of approximately 400 mm and 350 mm respectively, as we can see in Fig. (7) (a). Rotation q_5 around the y-axis decreases steadily and converges ultimately to its desired value.

5 CONCLUSION

We have presented a path-planning visual servoing strategy for cameras obeying the unified model, which includes conventional perspective cameras, fisheye cameras, and catadioptric cameras. This strategy consists of designing image trajectories for eye-in-hand robotic systems that allow one to reach a desired location while satisfying typical visual servoing constraints. To this end, the available image features are reprojected onto a virtual plane independently on the type of camera used to acquire them, a feasible image trajectory is computed through polynomial programming, and an IBVS controller is used to track such a trajectory. The proposed strategy has been illustrated and validated through experimental results with a fisheye camera mounted on a 6-DOF robot arm.

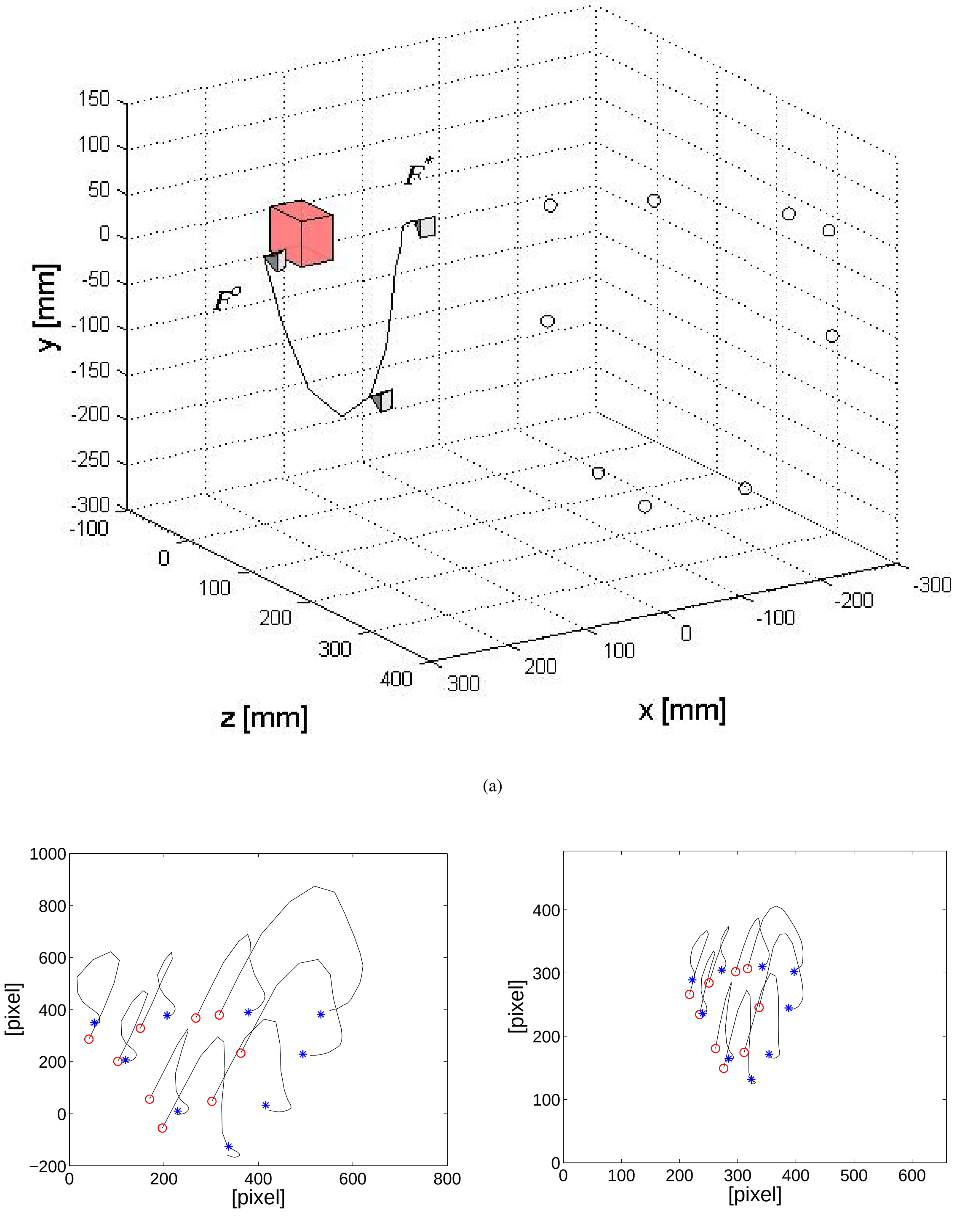
Future work will investigate the extension of the proposed strategy to complex objects where points are not available, and to different camera models.

Acknowledgment

The authors would like to thank the editors and reviewers for their valuable comments. This work was supported in part by the Research Grants Council of Hong Kong (Grants HKU711208E and HKU712808E).

REFERENCES

[1] C. Taylor and J. Ostrowski, “Robust vision-based pose control,” in *IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, 2000, pp. 2734–2740.

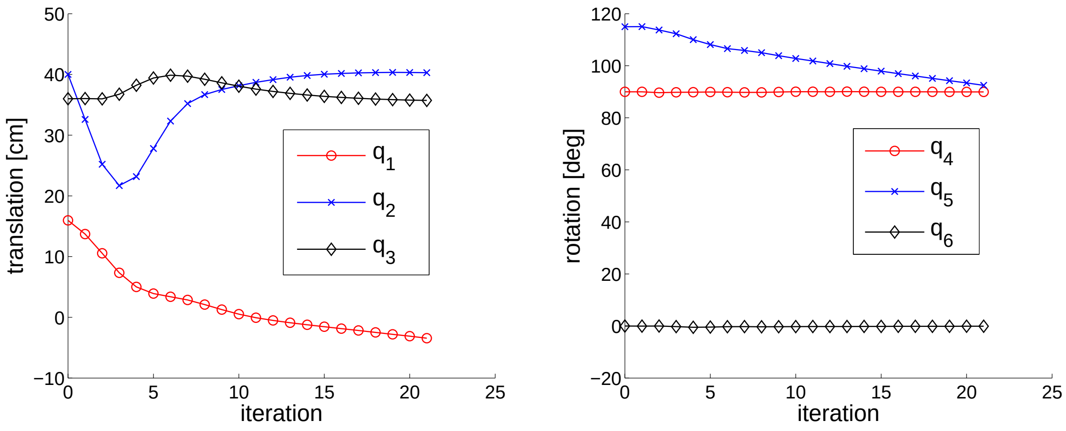


(a)

(b)

(c)

Figure 6: Experimental example 2. (a) 3D path. (b) Trajectories on the virtual plane. (c) Image trajectories.



(a)

(b)

Figure 7: Experimental example 2. (a) Robot translation evolution. (b) Robot rotation evolution.

[2] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proc. IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, 1991, pp. 2267–2272.

[3] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, D. Kriegman, G. . Hager, and A. Morse, Eds. LNCIS Series, No 237, Springer-Verlag, 1998, pp. 66–78.

[4] P. Y. Oh and P. K. Allen, "Visual servoing by partitioning degrees of freedom," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 1–17, 2001.

[5] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, April 1999.

[6] N. Gans and S. Hutchinson, "Stable visual servoing through hybrid switched-system control," *Robotics, IEEE Transactions on*, vol. 23, no. 3, pp. 530 –540, june 2007.

[7] N. Cowan, J. Weingarten, and D. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. on Robotics and Automation.*, vol. 18, no. 4, pp. 521–533, 2002.

[8] G. Chesi and A. Vicino, "Visual servoing for large camera displacements," *IEEE Trans. Robot.*, vol. 20, no. 4, 2004.

[9] G. Chesi, "Optimal object configurations for minimizing the positioning error in visual servoing," *IEEE Trans. on Robotics*, vol. 26, no. 3, pp. 584–589, 2010.

[10] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.

[11] G. Chesi, "Visual servoing path-planning via homogeneous forms and LMI optimizations," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 281–291, 2009.

[12] K. G. M. Kazemi and M. Mehrandezh, “Path-planning for visual servoing: A review and issues,” in *Visual Servoing via Advanced Numerical Methods*, G. Chesi and K. Hashimotos, Eds. Springer, 2010, pp. 189–207.

[13] F. Chaumette and S. Hutchinson, “Visual servo control, part I: basic approaches,” *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[14] ——, “Visual servo control, part II: advanced approaches,” *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[15] G. Chesi and K. Hashimoto, Eds., *Visual Servoing Via Advanced Numerical Methods*. Springer, 2010.

[16] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet, “A generic fisheye camera model for robotic application,” in *Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, 2007, pp. 1683–1688.

[17] O. Tahri, Y. Mezouar, F. Chaumette, and H. Araujo, “Visual servoing and pose estimation with cameras obeying the unified model,” in *Visual Servoing via Advanced Numerical Methods*, G. Chesi and K. Hashimotos, Eds. Springer, 2010, pp. 231–250.

[18] T. T. Shen, G. Chesi, and Y. S. Hung, “A visual servoing path-planning strategy for cameras obeying the unified model,” in *IEEE Multi-Conf. on Systems and Control*, Yokohama, Japan, September 2010, pp. 1795–1800.

[19] J. Weng, T. S. Huang, and N. Ahuja, “Motion and structure from two perspective views: algorithms, error analysis, and error estimation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 451–476, 1989.

[20] J. Craig, Ed., *Introduction To Robotics: Mechanics And Control*. Pearson Education, 2005.

[21] N. Mansard and F. Chaumette, “A new redundancy formalism for avoidance in visual servoing,” in *International Conference on Intelligent Robots and Systems*, 2005.

[22] M. Shahamiri and M. Jagersand, “Uncalibrated visual servoing using a biased newton method for on-line singularity detection and avoidance,” in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, aug. 2005, pp. 3953 – 3958.

[23] C. Mei and P. Rives, “Single view point omnidirectional camera calibration from planar grids,” in *IEEE International Conference on Robotics and Automation*, April 2007.