Following a Straight Line in Visual Servoing with Elliptical Projections

Tiantian Shen[1] and Graziano Chesi[2]
[1]*Department of Electronic Information Engineering, College of Polytechnic, Hunan Normal University, Changsha, China*
[2]*Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong*

Keywords: Elliptical Projection, Path Planning, Visual Servoing.

Abstract: The problem of visual servoing to reach the desired location keeping elliptical projections in the camera field of view (FOV) while following a straight line is considered. The proposed approach is representing the whole path with seven polynomials of a path abscise: variables in polynomial coefficients for translational path being zero to represent a minimum path length and for rotational part being adjustable satisfying the FOV limit. The planned elliptical trajectories are tracked by an image-based visual servoing (IBVS) controller. The proposed strategy is verified by a simulational case with a circle and a superposed point, where a traditional IBVS controller directs the camera a detour to the ground, the proposed approach however keeps straight the camera trajectory and also the circle visible. In addition, a six degrees of freedom (6-DoF) articulated arm mounted with a pinhole camera is used to validate the proposed method by taking three Christmas balls as the target.

1 INTRODUCTION

Visual servoing is a technique which uses visual information to control the robot moving to a desired location. Classical methods include image-based visual servoing (IBVS) (Hashimoto et al., 1991) and position-based visual servoing (PBVS) (Taylor and Ostrowski, 2000). They have well documented weaknesses and strengths (Chaumette, 1998b). In order to better satisfy constraints that arise in visual servoing, there appeared many other approaches: 2 1/2-D visual servoing (Malis et al., 1999), partition of the degrees of freedom (Oh and Allen, 2001), switched controllers (Gans and Hutchinson, 2007; Chesi et al., 2004), navigation functions (Cowan et al., 2002), path-planning techniques (Mezouar and Chaumette, 2002; Shen and Chesi, 2012a; Shen et al., 2013), omnidirectional vision systems (Fomena and Chaumette, 2008), invariant visual features from spherical projection (Tahri et al., 2013) and etc. See also the survey papers (Chaumette and Hutchinson, 2006; Chaumette and Hutchinson, 2007) and the book (Chesi and Hashimoto, 2010) for more details.

In addition to pixel coordinates of some representational points as visual features used in a controller, other features have also been explored including: image moments (Chaumette, 2004; Tahri and Chaumette, 2005; Fomena and Chaumette, 2008), luminance(Collewet and Marchand, 2010) and some invariant features computed from spherical projection (Tahri et al., 2013). These works allow solid objects that are more natural than a point, such as circle, sphere and cylinder to be considered as targets in visual servoing. High-level control strategies, such as path-planning techniques, are seldom considered in these works to enhance the system robustness by taking into account constraints like the camera field of view (FOV) limits, convergence in workspace and etc. This paper belongs to a series of works aiming at path-planning visual servoing based on image moments of some solid objects. Our previous work in (Shen and Chesi, 2012b) used spheres as the target to achieve multiple constraints including FOV limit of the sphere, occlusion avoidance among spheres and collision avoidance in workspace. Two simulation scenarios were considered: a sphere with two points and three spheres.

This paper focus on steering a straight camera path in the Cartesian space, while achieving convergence and camera FOV limit in visual servoing with elliptical projections (mainly circles). The proposed approach is representing the whole path with seven polynomials of a path abscise: variables in polynomial coefficients for translational path being zero to represent a minimum path length and for rotational part being adjustable satisfying the FOV limit. The planned elliptical trajectories are tracked by an image-based visual servoing controller. The proposed strat-

Shen, T. and Chesi, G.
Following a Straight Line in Visual Servoing with Elliptical Projections.
DOI: 10.5220/0005960400470056
In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016) - Volume 1*, pages 47-56
ISBN: 978-989-758-198-4

egy also applies to elliptical projections from spheres. It is verified by a simulational case with a circle and a superposed point, where a traditional IBVS controller directs the camera a detour to the ground, the proposed approach however keeps straight the camera trajectory and also the circle visible. At last, an experimental case with three Christmas balls as the target is used to validate again the proposed method.

The paper is organized as follows. Section II introduces the notation and elliptical projection of a circle. Section III presents the proposed strategy for following a straight line while keeping camera FOV of circular objects. Section IV shows simulation and experimental results. Lastly, Section V concludes the paper with some final remarks.

2 PRELIMINARIES

Let $\mathcal{R}$ denote the real number set, $\mathbf{I}_n$ the $n \times n$ identity matrix, $\mathbf{e}_i$ the i-th column of 3×3 identity matrix, $\mathbf{0}_n$ the $n \times 1$ null vector, $\mathbf{u} * \mathbf{v}$ the convolution of vectors $\mathbf{u}$ and $\mathbf{v}$, $[\mathbf{v}]_\times$ the skew-symmetric matrix of $\mathbf{v} \in \mathcal{R}^3$. Given two camera frames $F^\circ = \{\mathbf{R}, \mathbf{t}\}$ and $F^* = \{\mathbf{I}_3, \mathbf{0}_3\}$, the pose transformation from F° to F^* is expressed as $\{\mathbf{R}^\top, -\mathbf{R}^\top \mathbf{t}\}$. Suppose there is a 3D point expressed as $\mathbf{H} = [x, y, z]^\top$ in the camera frame F^*, then 3D coordinates of this point in the frame of F° is computed as $\mathbf{R}^\top(\mathbf{H} - \mathbf{t})$. Image projection of this point in camera frame F° is denoted as

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}^\top(\mathbf{H} - \mathbf{t}),$$

where $\mathbf{K} \in \mathcal{R}^{3\times 3}$ is the camera intrinsic parameters matrix:

$$\mathbf{K} = \begin{pmatrix} f_1 & 0 & u \\ 0 & f_2 & v \\ 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

In the above matrix, f_1 and f_2 are approximated values of the camera focal length, image plane of which has the boundary of $\zeta_x \times \zeta_y$ with $\zeta_x = 2u$ and $\zeta_x = 2v$. Elliptical projections are formed from either spheres or circles. It is assumed that in the camera frame of $F^* = \{\mathbf{I}_3, \mathbf{0}_3\}$, a circle is described by the intersection of a sphere and a plane (Chaumette, 1998a):

$$\begin{cases} (x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2 = r^2, \\ \alpha(x - x_o) + \beta(y - y_o) + \gamma(z - z_o) = 0. \end{cases} \tag{2}$$

The sphere is centered at $\mathbf{o} = [x_o, y_o, z_o]^\top$ with r as its radius and the plane is determined by the point $\mathbf{o}$ and a normal vector $[\alpha, \beta, \gamma]^\top$. The corresponding projection is in the form of an ellipse:

$$K_0 X^2 + K_1 Y^2 + 2K_2 XY + 2K_3 X + 2K_4 Y + K_5 = 0,$$

$$\text{with} \begin{cases} K_0 = a^2 \triangle + 1 - 2ax_0, \\ K_1 = b^2 \triangle + 1 - 2by_0, \\ K_2 = ab\triangle - bx_0 - ay_0, \\ K_3 = ac\triangle - cx_0 - az_0, \\ K_4 = bc\triangle - cy_0 - bz_0, \\ K_5 = c^2 \triangle + 1 - 2cz_0. \end{cases} \tag{3}$$

In the above function, a, b, c and $\triangle$ are given by:

$$\begin{cases} a = \alpha/\delta, b = \beta/\delta, c = \gamma/\delta, \\ \triangle = x_0^2 + y_0^2 + z_0^2 - r^2, \\ \delta = x_0\alpha + y_0\beta + z_0\gamma \\ 1/z = aX + bY + c. \end{cases} \tag{4}$$

In the image plane, the ellipse is displayed in Fig.1 with its centroid, lengths of major and minor radius and direction angle expressed in $K_i, i = 0, \ldots, 5$:

$$\begin{cases} \bar{X} = (K_1K_3 - K_2K_4)/(K_2^2 - K_0K_1), \\ \bar{Y} = (K_0K_4 - K_2K_3)/(K_2^2 - K_0K_1), \\ A^2 = \dfrac{2(K_0\bar{X}^2 + 2K_2\bar{X}\bar{Y} + K_1\bar{Y}^2 - K_5)}{K_0 + K_1 + \sqrt{(K_1 - K_0)^2 + 4K_2^2}}, \\ B^2 = \dfrac{2(K_0\bar{X}^2 + 2K_2\bar{X}\bar{Y} + K_1\bar{Y}^2 - K_5)}{K_0 + K_1 - \sqrt{(K_1 - K_0)^2 + 4K_2^2}}, \\ E = (K_1 - K_0 + \sqrt{(K_1 - K_0)^2 + 4K_2^2})/2K_2, \\ \varphi = \arctan(E). \end{cases} \tag{5}$$

Degenerating case when projection boils down into a segment will be avoided in the proposed path

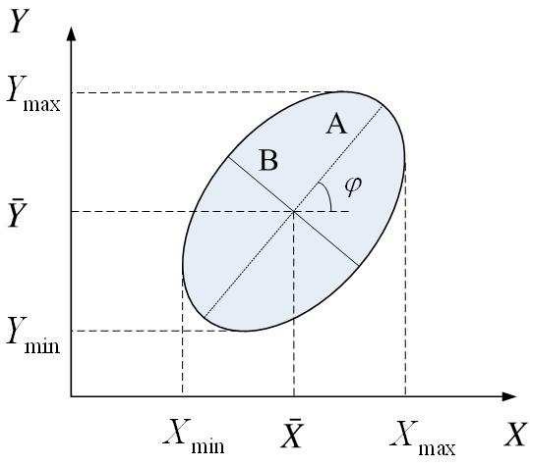


Figure 1: Elliptical projection.

planning algorithm, therefore not considered here. Extreme values of elliptical projection in Fig.1 is given by (Shen and Chesi, 2012b): $X_{max} = \bar{X} + \sqrt{\mu_{20}}$, $X_{min} = \bar{X} - \sqrt{\mu_{20}}$,

$Y_{max} = \bar{Y} + \sqrt{\mu_{02}}$, $Y_{min} = \bar{Y} - \sqrt{\mu_{02}}$, where μ_{ij} is the central image moments of the pertinent elliptical projection, whose centroid is calculated from raw image moments m_{ij} in the form of $\bar{X} = m_{10}/m_{00}$ and $\bar{Y} = m_{01}/m_{00}$ with m_{00} calculates the area of the ellipse.

Problem. The problem consists of steering an eye-in-hand robotic arm towards the desired location following a straight line of the camera in the Cartesian space, with the help of some elliptical projections. Convergence and FOV limit are to be imposed when features are mainly computed from image moments of elliptical projections.

3 PATH PLANNING

Before visual servoing, a straight path of the camera in the Cartesian space will be planned in advance. Planning of this straight path needs two end positions of the camera, this relative camera displacement is estimated via a virtual VS method based on image moments. However, it is not enough by only restraining in the Cartesian space, since corresponding image trajectories may go out of the image boundaries and fail later visual servoing applications. Polynomial minimization solves for the FOV problem and generates a satisfactory trajectory by adjusting the rotational path. The planned elliptical trajectories are then tracked by an IBVS controller in order to follow a straight camera path in the Cartesian space, at the same time without losing target features.

3.1 Two Ends of a Path

Requirement of a straight path in the Cartesian space and also camera FOV limits motivate a path-planning method that gives a satisfactory image trajectory to follow. Relative camera pose between the initial and the desired positions will serve as boundaries for subsequent path optimization. The target location and model are assumed to be known as a priori and two camera views of the target are given. From two views of the target and an approximated camera intrinsic parameters in (1), camera pose between F^* and F^o is estimated by virtually moving the camera from F^* to F^o with a traditional IBVS controller (Tahri et al., 2010):

$$\mathbf{T}_c(t) = -\lambda_1 \hat{\mathbf{L}}^+ (\mathbf{s}(t) - \mathbf{s}^*), \tag{6}$$

where $\mathbf{T}_c(t)$ is a camera velocity screw at time t, λ_1 is a positive gain, $\mathbf{s}(t)$ holds the current feature values and $\mathbf{s}^*$ the desired feature values, $\hat{\mathbf{L}}^+$ is the pseudo-inverse of the estimated interaction matrix corresponding to the selected features. Features of a circle is usually computed from image moments:

$$\mathbf{s} = [\bar{X}, \bar{Y}, \mu_{20}, \mu_{11}, \mu_{02}]^\top. \tag{7}$$

As specified in Section 2, $\bar{X}, \bar{Y}$ are pixel coordinates of the centroid of the elliptical projection. The remaining central image moments are related to ellipsoid parameters in the following way (Chaumette, 1998a):

$$\begin{aligned} \mu_{20} &= (A^2 + B^2E^2)/(1+E^2), \\ \mu_{11} &= E(A^2 - B^2)/(1+E^2), \\ \mu_{02} &= (A^2E^2 + B^2)/(1+E^2). \end{aligned} \tag{8}$$

The interaction matrix associated with the feature set in (7) is referred to the work in (Chaumette, 1998a). In order to reduce the chance of local minimum, we will add two more features, that are pixel coordinates of a point on the circle, to the feature set of a circle in (7). When spheres are taken as a target, we consider at least three spheres to reduce the possibility of nonsingular occurrence of the interaction matrix $\mathbf{L}$. Feature set for three spheres consists of nine values based on the computation of image moments:

$$\begin{aligned} \mathbf{s} = \Big[& \bar{X}_1, \bar{Y}_1, \tfrac{\mu_{02_1} + \mu_{20_1}}{2}, \bar{X}_2, \bar{Y}_2, \tfrac{\mu_{02_2} + \mu_{20_2}}{2}, \\ & \bar{X}_3, \bar{Y}_3, \tfrac{\mu_{02_3} + \mu_{20_3}}{2} \Big]^\top. \end{aligned} \tag{9}$$

The associated interaction matrix may be found in our previous work (Shen and Chesi, 2012b). The camera position is updated iteration by iteration by taking steps computed from velocity screw $\mathbf{T}_c(t)$ and time interval until feature error $|\mathbf{s}(t) - \mathbf{s}^*|$ is small enough. If no local minima problem exists, the ultimate camera position will be utilized in the subsequent polynomial parameterization and optimization of the camera path.

3.2 Polynomial Minimization

To describe the camera path with boundaries on both sides, we use a path abscise $w \in [0,1]$ with its value 0 implying the start of the path F^o, and value 1 meaning the end of the path F^*. Transition from F^o to F^* is developed from the results in Section 3.1 and denoted as $\{\mathbf{R}, \mathbf{t}\}$. Thus we have:

$$\begin{cases} \{\mathbf{R}(0), \mathbf{t}(0)\} = \{\mathbf{I}_3, \mathbf{0}_3\}, \\ \{\mathbf{R}(1), \mathbf{t}(1)\} = \{\mathbf{R}, \mathbf{t}\}. \end{cases} \tag{10}$$

Between the above two camera poses, camera path $\{\mathbf{R}(w), \mathbf{t}(w)\}$ is intended to satisfy the FOV limits

when $\mathbf{t}(w)$ is a straight line. With path abscise w changes from 0 to 1, camera translation $\mathbf{t}(w)$ linearly changes from $\mathbf{0}_3$ to $\mathbf{t}$, camera rotation $\mathbf{R}(w)$ changes from $\mathbf{I}_3$ to $\mathbf{R}$. In between these two ends, we use first-order polynomials in w to model translational path and second-order polynomials in w the rotational quaternions:

$$\begin{cases} \mathbf{q}(w) = \mathbf{U} \cdot [w^2, w, 1]^\top, \\ \mathbf{t}(w) = \mathbf{V} \cdot [w, 1]^\top, \end{cases} \tag{11}$$

with

$$\mathbf{q}(w) = \begin{bmatrix} \sin\dfrac{\theta(w)}{2}\mathbf{a}(w) \\ \cos\dfrac{\theta(w)}{2} \end{bmatrix}. \tag{12}$$

In quaternion representation, $\theta(w) \in (0, \pi)$ and $\mathbf{a}(w) \in \mathcal{R}^3$ are respectively rotation angle and axis of $\mathbf{R}(w)$ such that $\mathbf{R}(w) = e^{\theta(w)[\mathbf{a}(w)]_\times}$.

Boundaries defined in (10) will be satisfied by assigning the first and last columns in the coefficient matrices in (11) with the other entries variable:

$$\begin{cases} \mathbf{U} = [\mathbf{u}_1^\top, \mathbf{u}_2^\top, \mathbf{u}_3^\top, \mathbf{u}_4^\top]^\top \\ \quad = [\mathbf{q} - \mathbf{b}, \mathbf{b}, \mathbf{0}_3], \\ \mathbf{V} = [\mathbf{v}_1^\top, \mathbf{v}_2^\top, \mathbf{v}_3^\top]^\top \\ \quad = [\mathbf{t}, \mathbf{0}_3], \end{cases} \tag{13}$$

where $\mathbf{u}_i^\top \in \mathcal{R}^{1\times 3}$, $i = 1, \cdots, 4$ are the i-th row of matrix $\mathbf{U}$ and $\mathbf{v}_j^\top \in \mathcal{R}^{1\times 2}$, $j = 1, 2, 3$ the j-th row of matrix $\mathbf{V}$. In (13), vector $\mathbf{b}$ is variable and we assign it with an initial value of $\mathbf{0}_4$. With this initial value of $\mathbf{b}$, we can plot the parameterized path for the scenario in Fig 2 and then we have a straight path in the Cartesian space as shown in Fig. 3 (a). However, Fig. 3 (b) shows that the corresponding image trajectory goes out of the image boundary.

Therefore, we need to find an appropriate value of $\mathbf{b}$ in (13) that lets image trajectories fall within the image boundaries. For a circle, depth of the circle center is meant to be larger than the circle radius. In addition, extreme values of the elliptical projection that is drawn in Fig. 1 are restricted to being located within the image size:

$$\begin{cases} z_o - r > 0, \\ X_{max} < \dfrac{\zeta_x}{2f}, \\ X_{min} > -\dfrac{\zeta_x}{2f}, \\ Y_{max} < \dfrac{\zeta_y}{2f}, \\ Y_{min} > -\dfrac{\zeta_y}{2f}, \end{cases} \tag{14}$$

where X_{max}, Y_{max}, X_{min} and Y_{min} are computed from image moments of the sphere. They are expressed as:

$$\begin{aligned} X_{max} &= (K_2K_4 - K_1K_3 + \sqrt{G_2})/(K_0K_1 - K_2^2), \\ X_{min} &= (K_2K_4 - K_1K_3 - \sqrt{G_2})/(K_0K_1 - K_2^2), \\ G_2 &= (K_1K_3 - K_2K_4)^2 - (K_0K_1 - K_2^2)(K_1K_5 - K_4^2), \\ Y_{max} &= (K_2K_3 - K_0K_4 + \sqrt{G_3})/(K_0K_1 - K_2^2), \\ Y_{min} &= (K_2K_3 - K_0K_4 - \sqrt{G_3})/(K_0K_1 - K_2^2), \\ G_3 &= (K_0K_4 - K_2K_3)^2 - (K_0K_1 - K_2^2)(K_0K_5 - K_3^2). \end{aligned}$$

In order to realize conditions in (14), we wish to have $z_o - r$, X_{max}, Y_{max}, X_{min} and Y_{min} to be polynomials in the path abscise w. After path parameterization, in any camera frame of $\{\mathbf{R}(w), \mathbf{t}(w)\}$, we can assure that the circle center and the normal vector in (2) are polynomials in the path abscise. Let

$$[\mathbf{h}_1^\top, \mathbf{h}_2^\top, \mathbf{h}_3^\top]^\top = [\mathbf{0}_3, \mathbf{o}] - \mathbf{V}, \tag{15}$$

which indicates $\mathbf{h}_j^\top$, $j = 1, 2, 3$ the j-th row of matrix $[\mathbf{0}_3, \mathbf{o}] - \mathbf{V}$, where $\mathbf{o}$ is the circle center in (2) and $\mathbf{V}$ is coefficient matrix in (11). After camera displacement caused by varying w, polynomial coefficients of the circle center and the normal vector are computed as:

$$\begin{aligned} \mathbf{p}_{x_o} &= \mathbf{r}_{11} * \mathbf{h}_1 + \mathbf{r}_{21} * \mathbf{h}_2 + \mathbf{r}_{31} * \mathbf{h}_3, \\ \mathbf{p}_{y_o} &= \mathbf{r}_{12} * \mathbf{h}_1 + \mathbf{r}_{22} * \mathbf{h}_2 + \mathbf{r}_{32} * \mathbf{h}_3, \\ \mathbf{p}_{z_o} &= \mathbf{r}_{13} * \mathbf{h}_1 + \mathbf{r}_{23} * \mathbf{h}_2 + \mathbf{r}_{33} * \mathbf{h}_3, \\ \mathbf{p}_\alpha &= \alpha\mathbf{r}_{11} + \beta\mathbf{r}_{21} + \gamma\mathbf{r}_{31}, \\ \mathbf{p}_\beta &= \alpha\mathbf{r}_{12} + \beta\mathbf{r}_{22} + \gamma\mathbf{r}_{32}, \\ \mathbf{p}_\gamma &= \alpha\mathbf{r}_{13} + \beta\mathbf{r}_{23} + \gamma\mathbf{r}_{33}, \end{aligned} \tag{16}$$

where $\mathbf{p}_{x_o}, \mathbf{p}_{y_o}, \mathbf{p}_{z_o} \in \mathcal{R}^7$ and $\mathbf{p}_\alpha, \mathbf{p}_\beta, \mathbf{p}_\gamma \in \mathcal{R}^5$, $\mathbf{r}_{ij} \in \mathcal{R}^5$ are polynomial coefficients of the entry lie in the i-th row and the j-th column of the rotation matrix. Specifically, we have

$$\begin{aligned} \mathbf{r}_{11} &= \mathbf{u}_1 * \mathbf{u}_1 - \mathbf{u}_2 * \mathbf{u}_2 - \mathbf{u}_3 * \mathbf{u}_3 + \mathbf{u}_4 * \mathbf{u}_4, \\ \mathbf{r}_{12} &= 2(\mathbf{u}_1 * \mathbf{u}_2 - \mathbf{u}_3 * \mathbf{u}_4), \\ \mathbf{r}_{13} &= 2(\mathbf{u}_1 * \mathbf{u}_3 + \mathbf{u}_2 * \mathbf{u}_4), \\ \mathbf{r}_{21} &= 2(\mathbf{u}_1 * \mathbf{u}_2 + \mathbf{u}_3 * \mathbf{u}_4), \\ \mathbf{r}_{22} &= -\mathbf{u}_1 * \mathbf{u}_1 + \mathbf{u}_2 * \mathbf{u}_2 - \mathbf{u}_3 * \mathbf{u}_3 + \mathbf{u}_4 * \mathbf{u}_4, \\ \mathbf{r}_{23} &= 2(\mathbf{u}_2 * \mathbf{u}_3 - \mathbf{u}_1 * \mathbf{u}_4), \\ \mathbf{r}_{31} &= 2(\mathbf{u}_1 * \mathbf{u}_3 - \mathbf{u}_2 * \mathbf{u}_4), \\ \mathbf{r}_{32} &= 2(\mathbf{u}_2 * \mathbf{u}_3 + \mathbf{u}_1 * \mathbf{u}_4), \\ \mathbf{r}_{33} &= -\mathbf{u}_1 * \mathbf{u}_1 - \mathbf{u}_2 * \mathbf{u}_2 + \mathbf{u}_3 * \mathbf{u}_3 + \mathbf{u}_4 * \mathbf{u}_4. \end{aligned} \tag{17}$$

From equation (16), we can deduce that $z_0 - r$ in (14) is a six order polynomial in w with its coefficients to be

$$\mathbf{p}_{z_o - r} = [\mathbf{0}_6^\top, r] - \mathbf{p}_{z_o}. \tag{18}$$

Since extreme values X_{max}, Y_{max}, X_{min} and Y_{min} are functions of K_i, $i = 1, \cdots, 5$ as given in (3.2), this formulation will troubles polynomial parametrization of these values. In order to transform inequalities in (14) concerning these extreme values into polynomials, let

$$k_i = \delta^2 K_i, i = 1, \ldots, 5, \tag{19}$$

and we will have polynomial coefficients of all of k_i as follows:

$$\begin{aligned}
&\mathbf{p}_\triangle = \mathbf{p}_{x_o} * \mathbf{p}_{x_o} + \mathbf{p}_{y_o} * \mathbf{p}_{y_o} + \mathbf{p}_{z_o} * \mathbf{p}_{z_o} - [\mathbf{0}_{11}^\top, r^2],\\
&\mathbf{p}_\delta = \mathbf{p}_\alpha * \mathbf{p}_{x_o} + \mathbf{p}_\beta * \mathbf{p}_{y_o} + \mathbf{p}_\gamma * \mathbf{p}_{z_o},\\
&\mathbf{p}_{k_0} = \mathbf{p}_\alpha * \mathbf{p}_\alpha * \mathbf{p}_\triangle + \mathbf{p}_\delta * \mathbf{p}_\delta - 2\mathbf{p}_\alpha * \mathbf{p}_\delta * \mathbf{p}_{x_o},\\
&\mathbf{p}_{k_1} = \mathbf{p}_\beta * \mathbf{p}_\beta * \mathbf{p}_\triangle + \mathbf{p}_\delta * \mathbf{p}_\delta - 2\mathbf{p}_\beta * \mathbf{p}_\delta * \mathbf{p}_{y_o},\\
&\mathbf{p}_{k_2} = \mathbf{p}_\alpha * \mathbf{p}_\beta * \mathbf{p}_\triangle - \mathbf{p}_\beta * \mathbf{p}_\delta * \mathbf{p}_{x_o} - \mathbf{p}_\alpha * \mathbf{p}_\delta * \mathbf{p}_{y_o},\\
&\mathbf{p}_{k_3} = \mathbf{p}_\alpha * \mathbf{p}_\gamma * \mathbf{p}_\triangle - \mathbf{p}_\gamma * \mathbf{p}_\delta * \mathbf{p}_{x_o} - \mathbf{p}_\alpha * \mathbf{p}_\delta * \mathbf{p}_{z_o},\\
&\mathbf{p}_{k_4} = \mathbf{p}_\beta * \mathbf{p}_\gamma * \mathbf{p}_\triangle - \mathbf{p}_\gamma * \mathbf{p}_\delta * \mathbf{p}_{y_o} - \mathbf{p}_\beta * \mathbf{p}_\delta * \mathbf{p}_{z_o},\\
&\mathbf{p}_{k_5} = \mathbf{p}_\gamma * \mathbf{p}_\gamma * \mathbf{p}_\triangle + \mathbf{p}_\delta * \mathbf{p}_\delta - 2\mathbf{p}_\gamma * \mathbf{p}_\delta * \mathbf{p}_{z_o}.
\end{aligned} \tag{20}$$

All of the above $\mathbf{p}_{k_i} \in \mathcal{R}^{17}$, $i = 1, \cdots, 5$ are polynomial coefficients of k_i, $i = 1, \ldots, 5$. Let

$$\begin{aligned}
g_1 &= z_o - r,\\
g_8 &= k_0 k_1 - k_2^2,
\end{aligned} \tag{21}$$

we have first

$$\begin{aligned}
g_2 &= (k_1 k_3 - k_2 k_4)^2 - g_8(k_1 k_5 - k_4^2),\\
g_3 &= (k_0 k_4 - k_2 k_3)^2 - g_8(k_0 k_5 - k_3^2).
\end{aligned} \tag{22}$$

Positivity of g_2 and g_3 ensures that the maximum and minimum values of elliptical projection in the either X or Y direction are unequal. This constraint avoids the degenerating case: elliptical projection boils down to a segment. Image boundary limits are developed as follows:

$$\begin{aligned}
g_4 &= g_8 \left[\frac{\zeta_x}{2f} + k_2 k_4 - k_1 k_3\right]^2 - g_8 g_2 > 0,\\
g_5 &= g_1 \left[\frac{\zeta_x}{2f} - k_2 k_4 + k_1 k_3\right]^2 - g_8 g_2 > 0,\\
g_6 &= g_8 \left[\frac{\zeta_y}{2f} + k_2 k_3 - k_0 k_4\right]^2 - g_8 g_3 > 0,\\
g_7 &= g_1 \left[\frac{\zeta_y}{2f} - k_2 k_3 + k_0 k_4\right]^2 - g_8 g_3 > 0.
\end{aligned} \tag{23}$$

Since k_i, $i = 1, ..., 5$ are polynomials, therefore g_j, $j = 1, \cdots, 7$ are also polynomials in w. Their coefficients are computed from (18) and $\mathbf{p}_{k_i}$, $i = 1, \cdots, 5$ in (20). Take g_1 and g_2 for example, polynomial coefficients of g_1 is actually: $\mathbf{p}_{g_i} = \mathbf{p}_{z_o - r}$, and coefficients of g_2 is computed as:

$$\begin{aligned}
&\mathbf{p}_{k_1 k_3 - k_2 k_4} = \mathbf{p}_{k_1} * \mathbf{p}_{k_3} - \mathbf{p}_{k_2} * \mathbf{p}_{k_4},\\
&\mathbf{p}_{k_1 k_5 - k_4 k_4} = \mathbf{p}_{k_1} * \mathbf{p}_{k_5} - \mathbf{p}_{k_4} * \mathbf{p}_{k_4},\\
&\mathbf{p}_{g_8} = \mathbf{p}_{k_0} * \mathbf{p}_{k_1} - \mathbf{p}_{k_2} * \mathbf{p}_{k_2},\\
&\mathbf{p}_{g_2} = \mathbf{p}_{k_1 k_3 - k_2 k_4} * \mathbf{p}_{k_1 k_3 - k_2 k_4} - \mathbf{p}_{g_8} * \mathbf{p}_{k_1 k_5 - k_4 k_4}.
\end{aligned} \tag{24}$$

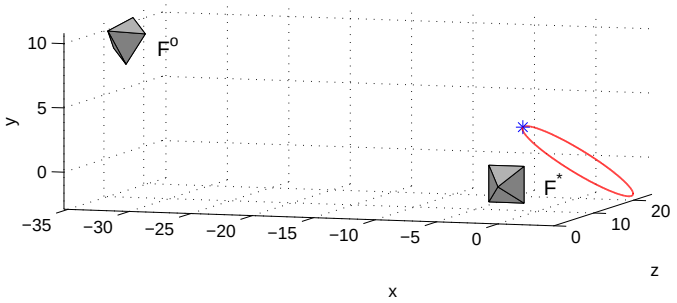
Similar computation process applies to the polynomial coefficients of g_j, $j = 3, \cdots, 7$. Provided the initial value of $\mathbf{b}$ in (13) is given, we derive values of $\mathbf{U}$ and $\mathbf{V}$ and bring them sequentially into (15), (17), (16), (18) and (20). Items in (20) are utilized in the computation of $\mathbf{p}_{g_2}$ in (24). In order to ensure the value of g_2 is positive when $w \in (0,1)$, we take the derivative of $\mathbf{p}_{g_2}$ and solve for the corresponding $w \in (0,1)$ that give zero derivatives. If such w exist, then a local minimum of g_j can be found at such w values. Take all of the positivity requirements of g_j, $j = 1, \cdots, 7$ together, we first find local minimums for each g_j, $j = 1, \cdots, 7$, and again the minimum of all of these local minimums and denote it as g^*:

$$\begin{aligned}
g^* &= \text{minimum}_{j=1}^{7}(\text{minimum}_{w\in(0,1)}(g_j)),\\
\mathbf{b}^* &= \min_{\mathbf{b}}(-g^*).
\end{aligned} \tag{25}$$

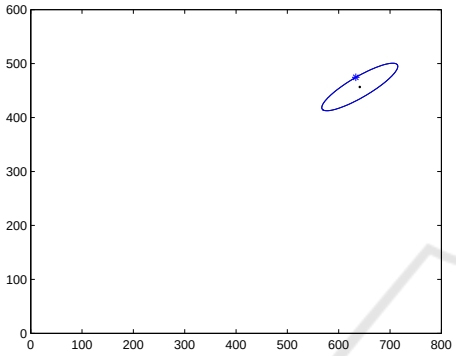
No action will be taken if the value of g^* is positive, otherwise a minimization of $-g^*$ will be conducted until it converts its sign. Take a synthetic scene of a circle as an example, Fig. 2 (a) consists of two camera positions and a circle as a target. Fig. 2 (b)-(c) are two views of the target. Relative camera pose between F^* and F^o is estimated using a virtual VS method demonstrated in Section 3.1. A straight path is then planned between F^* and F^o, as shown in Fig. 3 (c). In this scenario, path planning with initial value of $\mathbf{b}$, that is $\mathbf{0}_3$, will cause the image boundary violated though camera trajectory in the Cartesian space is a straight line, see Fig. 3 (d). This situation indicates that polynomial curve of g_7 will across below the horizontal axis over $w \in (0,1)$, and therefore g^* is negative. We search for appropriate value of $\mathbf{b}$ by minimizing $-g^*$ with MATLAB tool till $g^* > 0$. Once the minimization result is obtained, we denote it as $\mathbf{b}^*$ and bring it into (13). Till this end, a planned path that converges to the desired camera location following a straight line while satisfying the camera FOV limit is found, see Fig. 3 (e)-(f) as an example.

3.3 Tracking Image Trajectories

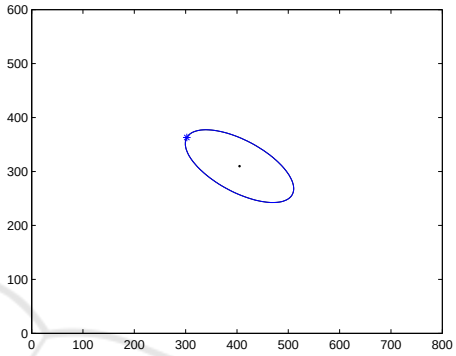
The robot manipulator is going to be servoed by tracking an well planned image trajectory. We substitute the desired feature value $\mathbf{s}^*$ in (6) with the planned feature value at time t, which is denoted as $\mathbf{s}_p(w_t)$



(a)



(b)



(c)

Figure 2: Scenery with a circle and a point on the circle. (a) Scenario (b) Camera view in F^o. (c) Camera view in F^*.

with $w_t = 1 - e^{-t\lambda_2}$:

$$\mathbf{T}_c = -\lambda_1 \hat{\mathbf{L}}^+ (\mathbf{s}(t) - \mathbf{s}_p(w_t)) + \lambda_3 \hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}_p(w_t)}{\partial t}, \quad (26)$$

The term $\lambda_3 \hat{\mathbf{L}}^+ \partial \mathbf{s}_p(w_t)/\partial t$ allows to compensate the tracking error (Mezouar and Chaumette, 2002). The partial derivative of the planned feature set is computed as:

$$\frac{\partial \mathbf{s}_p(w_t)}{\partial t} = \frac{\mathbf{s}_p(w_{t+\Delta t}) - \mathbf{s}_p(w_t)}{\Delta t}. \quad (27)$$

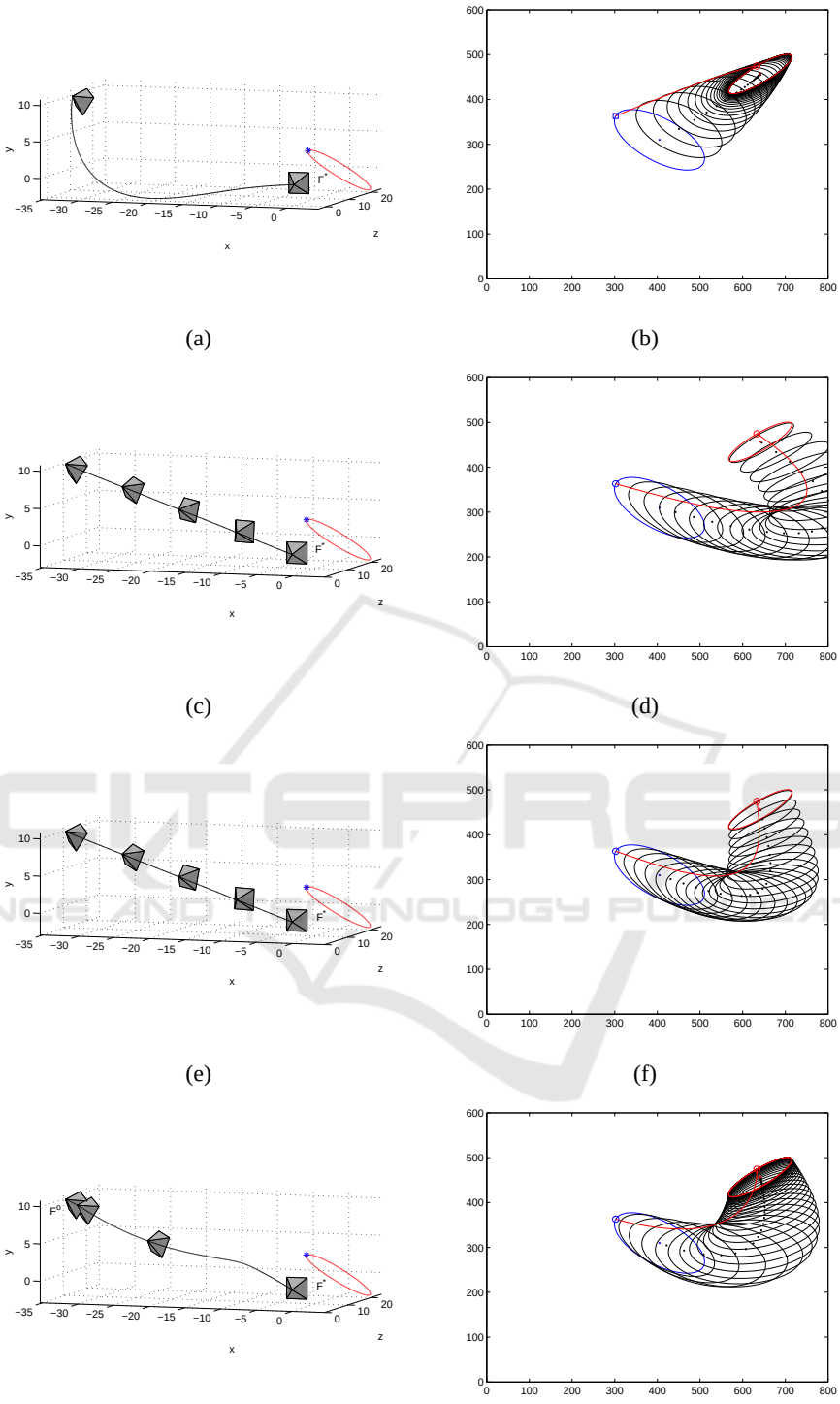
4 EXAMPLES

The first example aims to follow a straight line in the Cartesian space while keeping camera FOV of a circle. Synthetic scene is generated using MATLAB for the first example. The second example deals with three Christmas balls. The algorithm keeps visibility of all of the three balls in VS process while following a straight line.

4.1 Simulation with a Circle

The scenario for path planning with a circle is illustrated in Fig.2 (a). In the desired camera frame of $F^* = \{\mathbf{I}_3, \mathbf{0}_3\}$, model parameters of the circle are given as $[x_o, y_o, z_o] = [0,0,20]^\top$, $[\alpha, \beta, \gamma] = [0,0,1]^\top$ and $r = 5$mm. A point on the circle is also used to contribute features, shown as a star mark in Fig.2 (a). The initial and the desired camera views of the target are respectively shown in Fig.2 (b-c). From these two views, we extract features from elliptical projections of the circle and then utilize these features to complete planning a straight path from the initial camera frame $F^o = \{\mathbf{e}^{[\rho]\times}, [-35,10,10]^\top \text{mm}\}$ with $\rho = [\pi/12, \pi/4, -\pi/6]$ to the desired camera frame F^*. Along this straight path, the elliptical projections of the circle are expected to be always kept within the camera view. Intrinsic parameters of the camera are approximated with $f_1 = 456$, $f_2 = 448$, $u = 403$, $v = 301$ in (1).

Relative camera pose between F^* and F^o is estimated using a virtual VS method based on the selected features. Fig. 3 (a-b) shows the camera trajectory and image trajectories generated using virtual VS, where the camera trajectory is extremely close to the ground. In order to achieve the shortest travel length, a straight path in the Cartesian space is planned in Fig. 3 (c). With initial coefficient values in the polynomial parametrization, elliptical projections of the circle will go beyond the image boundary and fail a real visual servoing application, see Fig. 3 (d).



(a) (b)

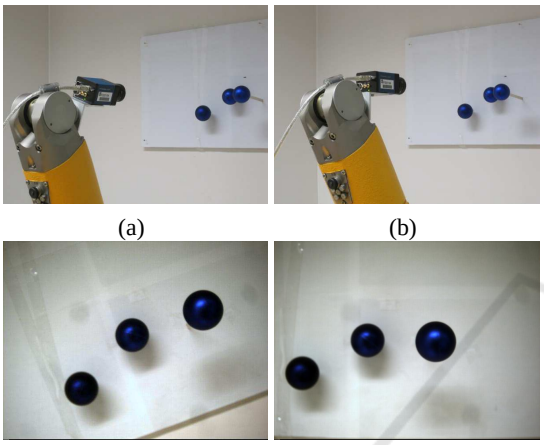
(c) (d)

(e) (f)

(g) (h)

Figure 3: Path planning with a circle and a point on the circle. (a) Camera trajectory in virtual IBVS. (b) Image trajectories in virtual IBVS. (c) Planned path in workspace and camera postures without optimization. (d) Image trajectories without optimization. (e) Planned path and camera postures in workspace. (f) Planned image trajectories. (g) Trajectory in the Cartesian space generated when applying an IBVS controller. (h) Image trajectories generated when applying an IBVS controller.

Therefore, minimization of $-g^*$ in (25) is performed to make an adjustment, after which a satisfactory image trajectory is generated as shown in Fig. 3 (f). The planned image trajectory is brought into an IBVS controller in (26), where positive gains λ_1, λ_2 and λ_3 are respectively taken as 0.015, 0.01 and 0.4. An instant camera velocity screw $\mathbf{T}_c$ is produced and used to direct the camera moving towards the desired location by iterations. Data recorded at each iteration are plotted in Fig. 3 (g)-(h).



(a) (b) (c) (d)

Figure 4: Experiment with three Christmas balls. (a) Initial robot/camera pose. (b) Desired robot/camera pose. (c) Initial view. (d) Desired View.

4.2 Experiment with Christmas Balls

This is a very simple experimental example validating the proposed strategy. The robot used for the experiment is a Staubli RX60 6-DoF articulated arm, on the robot end-effector mounted a video camera. The camera is calibrated with its intrinsic parameters as follows:

$$\mathbf{K} = \begin{pmatrix} 851.76868 & 0 & 329.00000 \\ 0 & 851.76868 & 246.00000 \\ 0 & 0 & 1 \end{pmatrix}. \tag{28}$$

The target consists of three Christmas balls with approximately the same radius of 20 mm. Positions of these three ball centers in the desired camera frame F^* are approximately

$$\mathbf{O}_1 = [25.0183, -0.0005, 344.8407]^\top \quad \text{mm},$$
$$\mathbf{O}_2 = [-48.6174, -2.4141, 412.1984]^\top \quad \text{mm},$$
$$\mathbf{O}_3 = [-128.5265, -39.0117, 414.3849]^\top \quad \text{mm}.$$

Fig. 4 shows the two configurations and corresponding camera views. Displacement between the desired and the initial camera poses is estimated by virtual visual servoing:

$$\mathbf{R} = \begin{pmatrix} 0.9399 & 0.3415 & -0.0000 \\ -0.3415 & 0.9399 & 0.0001 \\ 0.0001 & -0.0001 & 1.0000 \end{pmatrix},$$

$$\mathbf{t} = (-44.2772, -0.0353, -0.0020)^\top \quad \text{mm}.$$

Trajectories generated in virtual VS process without path planning are plotted in Fig. 5 (a). In virtual VS, image trajectories of sphere centers are straight lines, however camera trajectory is not a straight one. After path planning and applying an IBVS controller to follow the planned path, real camera trajectory and image trajectories of the target are displayed in Fig. 5 (b)-(c), they are different from the ones generated in virtual VS process. Iterations of camera translation and rotation are recorded and plotted in Fig. 5 (d)-(e), which show the convergence in both camera translation and rotation. It is assumed that iteration number in real-time step-by-step moving robotic application is denoted as t, trajectories $\mathbf{q}_R(t) = [q_4(t), q_5(t), q_6(t)]^\top$ in Fig. 5 (e) are Cayley representation (Craig, 2005) of the associated rotation matrix $\mathbf{R}(t)$:

$$[\mathbf{q}_R(t)]_\times = (\mathbf{R}(t) - \mathbf{I}_3)(\mathbf{R}(t) + \mathbf{I}_3)^{-1}. \tag{29}$$

Fluctuations exist in the iterations mainly due to coarse feature extraction and the value of tracking gain used for computing the next arriving position in VS process for the step-by-step moving robot.

5 CONCLUSIONS

This paper proposes a straight path planning approach for visual servoing with elliptical projections. Circles and spheres may project into traditional cameras as ellipsoids. Constrains with spheres are considered in our previous work, this paper mainly solves for problems with circular objects, specifically the convergence and FOV limit while following a straight line in the Cartesian space. Simulation with a circle and experiment with balls validate the proposed path planning approach. For future work, the impact of image noises imported from the area-based feature extraction on the precision of pose estimation and the subsequent path planning needs to be taken case of. More complicated objects are going to be explored, such as a cup to be seen as a group of two lines or two circles.

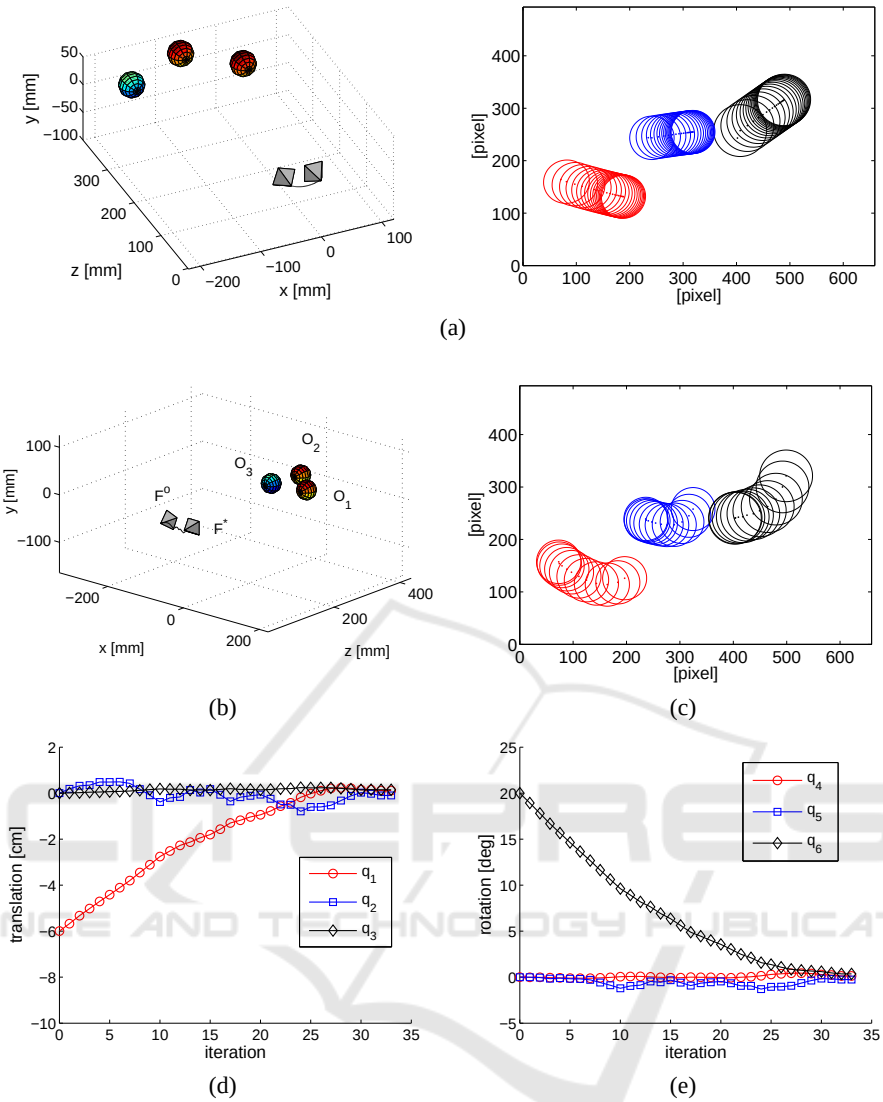


Figure 5: Experiment with three Christmas balls. (a) Virtual Camera path. (b) Virtual image trajectories. (c) Real camera path. (d) Real image trajectories. (e) Camera translation evolution. (f) Camera rotation evolution.

REFERENCES

Chaumette, F. (1998a). *De la perception à l'action: l'asservissement visuel; de l'action à la perception: la vision active*. Habilitation à diriger les recherches, Université de Rennes 1.

Chaumette, F. (1998b). Potential problems of stability and convergence in image-based and position-based visual servoing. In Kriegman, D., Hager, G., and Morse, A. S., editors, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag.

Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4):713–723.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control, part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90.

Chaumette, F. and Hutchinson, S. (2007). Visual servo control, part II: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118.

Chesi, G. and Hashimoto, K., editors (2010). *Visual Servoing via Advanced Numerical Methods*. Springer-Verlag.

Chesi, G., Hashimoto, K., Prattichizzo, D., and Vicino, A. (2004). Keeping features in the field of view in eye-in-hand visual servoing: A switching approach. *IEEE Transactions on Robotics*, 20(5):908–914.

Collewet, C. and Marchand, E. (2010). Luminance: a new visual feature for visual servoing. In Chesi, G. and Hashimotos, K., editors, *Visual Servoing via Ad-*

vanced Numerical Methods, pages 71–90. LNCIS Series, No 401, Springer-Verlag.

Cowan, N., Weingarten, J., and Koditschek, D. (2002). Visual servoing via navigation functions. *IEEE Transactions on Robotics and Automation*, 18(4):521–533.

Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson Education, 3rd edition.

Fomena, R. T. and Chaumette, F. (2008). Visual servoing from two special compounds of features using a spherical projection model. In *IROS'08, 21st IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3040–3045, Nice, France.

Gans, N. and Hutchinson, S. (2007). Stable visual servoing through hybrid switched-system control. *IEEE Transactions on Robotics*, 23(3):530–540.

Hashimoto, K., Kimoto, T., Ebine, T., and Kimura, H. (1991). Manipulator control with image-based visual servo. In *ICRA'91, 8th IEEE International Conference on Robotics and Automation*, pages 2267–2272, San Francisco, CA.

Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250.

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

Oh, P. Y. and Allen, P. K. (2001). Visual servoing by partitioning degrees of freedom. *IEEE Transactions on Robotics and Automation*, 17(1):1–17.

Shen, T. and Chesi, G. (2012a). Visual servoing path-planning for cameras obeying the unified model. *Advanced Robotics*, 26(8–9):843–860.

Shen, T. and Chesi, G. (2012b). Visual servoing path-planning with spheres. In *ICINCO'12, 9th International Conference on Informatics in Control, Automation and Robotics*, pages 22–30, Rome, Italy.

Shen, T., Radmard, S., Chan, A., Croft, E. A., and Chesi, G. (2013). Motion planning from demonstrations and polynomial optimization for visual servoing applications. In *IROS'13, 26th IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 578–583, Tokyo Big Sight, Japan.

Tahri, O., Araujo, H., Chaumette, F., and Mezouar, Y. (2013). Robust image-based visual servoing using invariant visual information. *Robotics and Autonomous Systems*, 61(12):1588–1600.

Tahri, O. and Chaumette, F. (2005). Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics*, 21(6):1116–1127.

Tahri, O., Mezouar, Y., Chaumette, F., and Araujo, H. (2010). Visual servoing and pose estimation with cameras obeying the unified model. In Chesi, G. and Hashimotos, K., editors, *Visual Servoing via Advanced Numerical Methods*, pages 231–252. LNCIS Series, No 401, Springer-Verlag.

Taylor, C. and Ostrowski, J. (2000). Robust vision-based pose control. In *ICRA'00, 17th IEEE International Conference on Robotics and Automation*, pages 2734–2740, San Francisco, CA.