

Visual Servoing Path-planning with Spheres

Tiantian Shen and Graziano Chesi

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

Keywords: Sphere, Path-planning, Visual Servoing.

Abstract: This paper proposes a path-planning approach for visual servoing in the case where the observed object features are points and spheres. Two main situations are considered. In the first situation, it is supposed that at least two points and at least a sphere are observed. In the second situation, it is supposed that at least three spheres are observed. The problem consists of planning a trajectory in order to ensure the convergence of the robot end-point to the desired location while satisfying visibility and workspace constraints, in particular including occlusion and collision avoidance. A solution based on polynomial parametrizations is proposed in order to determine a feasible path in the 3D space, and such a path is then followed by tracking its image projection through image-based visual servoing. Some simulation results illustrate the proposed approach.

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, combination of IBVS and PBVS is explored such as 2 1/2-D visual servoing (Malis et al., 1999), partition of the degrees of freedom (Oh and Allen, 2001) and switched controllers (Gans and Hutchinson, 2007; Chesi et al., 2004). Other approaches include: navigation functions (Cowan et al., 2002), circular-like trajectories for global convergence (Chesi and Vicino, 2004), path-planning techniques (Mezouar and Chaumette, 2002a; Chesi and Hung, 2007; Chesi, 2009; Shen and Chesi, 2012), omnidirectional vision systems (Tatsambon Fomena and Chaumette, 2008), 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 most of the literature, pixel coordinates of representational points are the dominant visual features used in a controller. On the other hand, other features more intuitive than points, such as image moments (Chaumette, 2004; Tahri and Chaumette, 2005; Tatsambon Fomena and Chaumette, 2008) and luminance (Collewet and Marchand, 2010), have been

explored in a small part of the literature. Among these works, solid objects that are more natural than a point, such as circle, sphere and cylinder (Chaumette, 1998a), are considered as targets in visual servoing. In particular, in the work of (Tatsambon Fomena and Chaumette, 2008), a set of three-dimensional (3-D) features are computed from image moments of a sphere and used in a classical control law that is proved to be globally asymptotically stable in the presence of modeling errors and locally asymptotically stable in the presence of calibration errors. These 3-D features are structured through spherical projection of the sphere, and therefore they are applicable to omnidirectional vision systems. In this reference, however, high-level control strategies, such as path-planning techniques, are not considered to take into account constraints.

Though omnidirectional vision systems are not the concern in this paper like what is considered in reference (Tatsambon Fomena and Chaumette, 2008), however, sphere is our common interest as a target. This paper aims to use path-planning techniques to achieve a variety of constraints that arise in visual servoing from spheres. A sphere may provide at most three independent features, and therefore we combine the sphere with at least two points or at least two other spheres to constitute the whole target. Consequently, two main situations are considered in this paper: a sphere with two points and three spheres. For each of the cases, constraints that are expected to be satisfied include field-of-view limit of the whole target (which contains at least one sphere), occlusion avoidance

among all the entities that consist of the target, and collision avoidance between the camera and an given obstacle in workspace. A camera path that meets all these constraints is planned based on polynomial parametrizations, and such a path is then followed by tracking its image projection through image-based visual servoing. Some simulation results illustrate the proposed approach.

The paper is organized as follows. Section II introduces the notation and image moments of a sphere. Section III presents the proposed strategy for visual servoing path-planning with spheres. Section IV shows some simulation results. Lastly, Section V concludes the paper with some final remarks.

2 PRELIMINARIES

We denote by \mathcal{R} the real number set, $\mathbf{I}_{n \times n}$ 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, $[\mathbf{v}]_{\times}$ the skew-symmetric matrix of $\mathbf{v} \in \mathcal{R}^3$

2.1 Frame Transformation

Given two camera frames $F_1 = \{\mathbf{R}_1, \mathbf{t}_1\}$ and $F_2 = \{\mathbf{R}_2, \mathbf{t}_2\}$ as shown in Fig. 1, the pose transformation from F_1 to F_2 is expressed as $\{\mathbf{R}, \mathbf{t}\}$:

$$\begin{cases} \mathbf{R} = \mathbf{R}_1^{\top} \mathbf{R}_2 \\ \mathbf{t} = \mathbf{R}_1^{\top} (\mathbf{t}_2 - \mathbf{t}_1). \end{cases} \quad (1)$$

Suppose in the camera frame F_1 there is a 3D point with its coordinates expressed as \mathbf{H} , then the coordinates of \mathbf{H} in the camera frame of F_2 is $\mathbf{R}^{\top} (\mathbf{H} - \mathbf{t})$. Image projection of this point in F_2 is denoted as $[X, Y, 1]^{\top} = \mathbf{K} \mathbf{R}^{\top} (\mathbf{H} - \mathbf{t})$, where $\mathbf{K} \in \mathcal{R}^{3 \times 3}$ is the camera intrinsic parameters matrix.

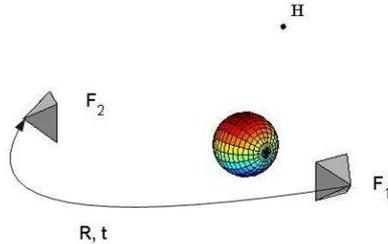


Figure 1: Transformation between two camera frames.

2.2 Projection of a Sphere

In particular, when the target contains a sphere with $\mathbf{o} = [x_o, y_o, z_o]^{\top}$ as its center coordinates expressed in the camera frame and r as its radius:

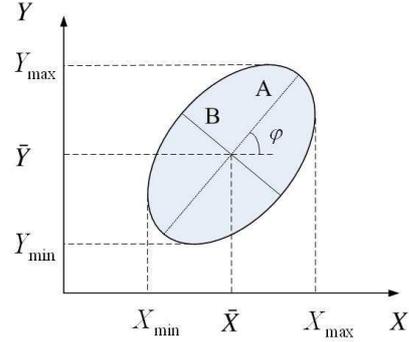


Figure 2: Projection of a sphere on the image plane.

$$(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2 = r^2, \quad (2)$$

image projection of the sphere in the same camera frame is in the form of an ellipse, the shape of which consists of a set of points:

$$\begin{cases} X(\phi) = \bar{X} + A \cos(\phi) \cos(\phi) - B \sin(\phi) \sin(\phi), \\ Y(\phi) = \bar{Y} + A \cos(\phi) \sin(\phi) + B \sin(\phi) \cos(\phi), \end{cases} \quad (3)$$

where $\phi \in (0, 2\pi]$, (\bar{X}, \bar{Y}) is the centroid of the ellipse, A and B are half values of the major and minor diameters, and ϕ describes the angle between the X-axis and the major axis of the ellipse as shown in Fig. 2. We induce the following relationship from results of (Chaumette, 1998a), assuming pixels in the image are in the shape of squares and f is the focal length of the camera:

$$\begin{cases} (\bar{X}, \bar{Y}, 1)^{\top} = \mathbf{K} \left(\frac{x_o z_o}{z_o^2 - r^2}, \frac{y_o z_o}{z_o^2 - r^2}, 1 \right)^{\top}, \\ A = f \sqrt{r^2 (x_o^2 + y_o^2 + z_o^2 - r^2) / (z_o^2 - r^2)}, \\ B = f \sqrt{r^2 / (z_o^2 - r^2)}, \\ \phi = \arccos(y_o / x_o). \end{cases} \quad (4)$$

Suppose that $I(X, Y)$ are pixel intensities of the ellipse area in a grey-scale image, raw image moments m_{ij} and central image moments μ_{ij} of the pertinent sphere are defined as:

$$m_{ij} = \sum_X \sum_Y X^i Y^j I(X, Y), \quad (5)$$

$$\mu_{ij} = \sum_X \sum_Y (X - \bar{X})^i (Y - \bar{Y})^j I(X, Y), \quad (6)$$

where m_{00} calculates the area of the ellipse, $\bar{X} = m_{10}/m_{00}$ and $\bar{Y} = m_{01}/m_{00}$ are the components of the centroid.

Problem. The problem consists of planning a trajectory for visual servoing from spheres in order to ensure the convergence of the robot end-point to the desired location while satisfying a lot of constraints. These constraints will include field-of-view limit of

the whole sphere instead of just a few representational points, collision avoidance between the camera and obstacles in workspace, and prevention of target self-occlusion when more than one entities are combined as a target.

3 PROPOSED STRATEGY

A sphere will provide at most three independent features. Therefore, we combine a sphere with at least two points or at least two other spheres to serve as a target. For each of the two cases, camera pose is estimated via a virtual VS method and later used as boundary for path-planning in workspace.

Between the estimated relative camera pose, a lot of constraints will be considered including camera field-of-view of the target, target-self occlusion avoidance, and collision avoidance between the camera and an given obstacle. To fulfill these constraints, trajectories of camera position and orientation are represented by polynomials in a common path parameter. We will search out the appropriate coefficients of these polynomials to determine a feasible path in the 3D space.

3.1 Pose Estimation

Multiple constraints satisfaction motivates an off-line path-planning method. The performance of path-planning in workspace greatly relies on the results of camera pose estimation. In this paper, we assume that intrinsic parameters of the camera are known a priori with calibration errors, and also the target position and model have already been given. With two views of the target, relative camera pose may be estimated using a virtual VS method (Tahri et al., 2010) that moves virtually the camera from F^* to F^o with instant camera velocities as follows:

$$\mathbf{T}_c(t) = -\lambda_1 \hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}^*), \quad (7)$$

where $\mathbf{T}_c(t) = [\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z, \boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z]^\top$ describes camera velocities in translation and rotation at time t , which decrease along with the falling trends of $|\mathbf{s}(t) - \mathbf{s}^*|$. $\mathbf{s}(t)$ denotes current features at time t , \mathbf{s}^* are the desired values of these selected features, $\hat{\mathbf{L}}^+$ is the pseudo-inverse of the estimated interaction matrix corresponding to the selected features. When the target consists of a sphere and two points, features will be selected as:

$$\mathbf{s} = \left[\bar{X}_1, \bar{Y}_1, \frac{\mu_{021} + \mu_{201}}{2}, X_2, Y_2, X_3, Y_3 \right]^\top, \quad (8)$$

where the first three items are composed of image moments of a sphere, X_j, Y_j are pixel coordinates of

image projection of the j -th point. The selection of $(\mu_{02} + \mu_{20})/2$ as a feature refers to the work in (Chaumette, 1998a).

The other case considered in this paper is a combination of at least three separate spheres. We give the feature set for three spheres:

$$\mathbf{s} = \left[\bar{X}_1, \bar{Y}_1, \frac{\mu_{021} + \mu_{201}}{2}, \bar{X}_2, \bar{Y}_2, \frac{\mu_{022} + \mu_{202}}{2}, \bar{X}_3, \bar{Y}_3, \frac{\mu_{023} + \mu_{203}}{2} \right]^\top. \quad (9)$$

If the target is composed of only two spheres, the chance of reaching local minima will be great in the process of visual servoing. In any of the previous mentioned circumstances, relative camera pose will be estimated by moving the camera in accordance with velocities obtained in (7) until the maximum value of vector $|\mathbf{s}(t) - \mathbf{s}^*|$ is smaller than a threshold. It is realized by the following iterations with initial values: $\mathbf{t} = \mathbf{0}_3, \mathbf{r} = \mathbf{0}_3, \mathbf{M} = \mathbf{I}_{3 \times 3}$.

$$\mathbf{t} \leftarrow \mathbf{t} + e^{[\mathbf{r}]^\times} [\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z]^\top \Delta t, \quad (10)$$

$$\mathbf{r} \leftarrow \mathbf{r} + \mathbf{M} [\boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z]^\top \Delta t, \quad (11)$$

$$\rho = \|\mathbf{r}\|, \quad \delta = -\mathbf{r}/\|\mathbf{r}\|, \quad (12)$$

$$\mathbf{M} = \mathbf{I}_{3 \times 3} - \frac{\rho}{2} [\delta]_\times + \frac{(\rho/2)^2}{\sin^2(\rho/2)} \left(1 - \frac{\sin \rho}{\rho} \right) [\delta]_\times^2. \quad (13)$$

At last, the estimated camera pose has the value of $\{e^{[\mathbf{r}]^\times}, \mathbf{t}\}$ when the iteration stops. It is expected that no local minima occurs in image space by comparing the estimated result with the ground truth. We use a simulation example to demonstrate the pose estimation method, where the target consists of three separate spheres as depicted in Fig. (3) (a). Radii and positions of these spheres are supposed to be known. Fig. (3) (b)-(c) are two views of the target obtained at different camera poses F^o and F^* . Relative camera pose between F^* and F^o is estimated using a virtual VS method with Fig. (3) (d) showing the virtual camera path generated by iteration (10)-(13) and Fig. (3) (e) displaying the associated image trajectories. It is noticed that there are two spheres occluded with each other in the virtual servoing process, the situation of which will fails a real VS task. Therefore, occlusion among these spheres must be prevented by a path-planning method, and it is also expected at the same time that all of these spheres are kept in the field-of-view of the camera. When an obstacle blocks the way of the camera, collision between them is intended to be avoided while satisfying simultaneously the other constraints.

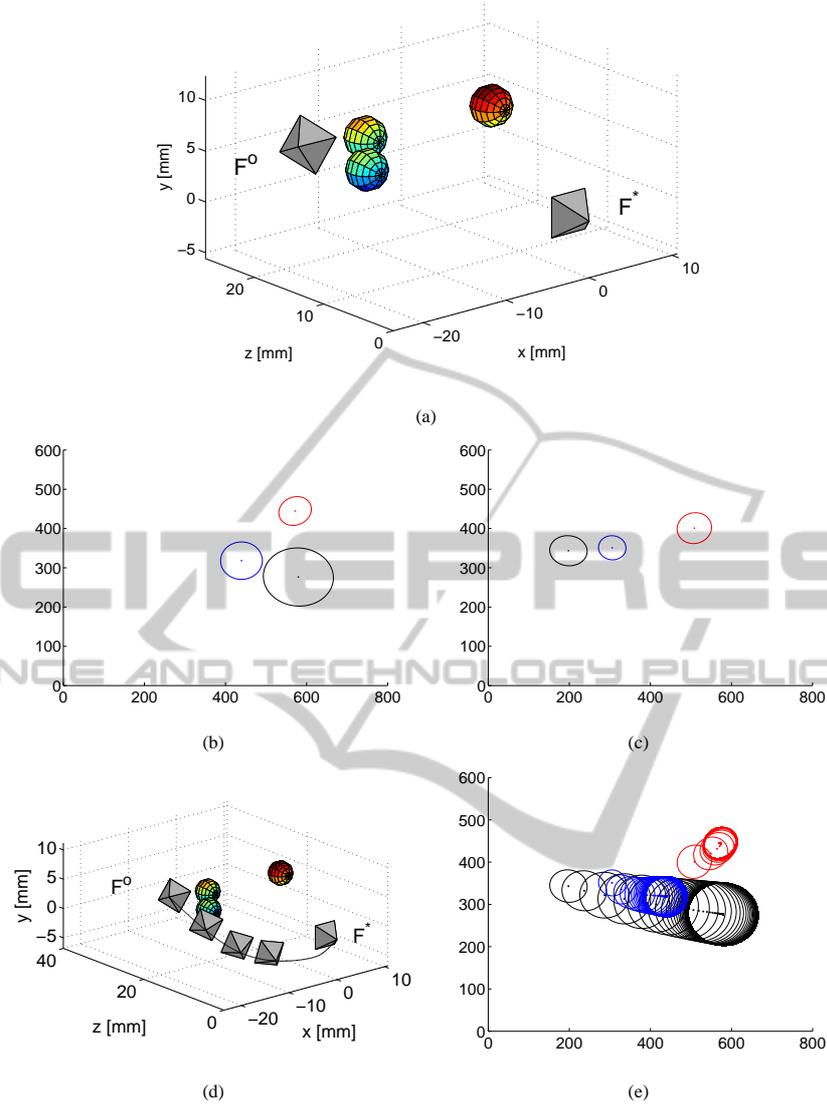


Figure 3: Pose estimation with three spheres. (a) Scenario. (b) Camera view in F^o . (c) Camera view in F^* . (d) Virtual camera path. (e) Virtual image trajectories.

3.2 Constraints

For all the above mentioned constraints, priority will be given to the depth of the target, which is required to be positive all the way along the camera path. Second, collision in workspace will be prevented by adjusting mainly the trajectories of camera translation. Third, field-of-view limit is going to be met by restraining projection of the target within the image size. When the target consists of two or more separate entities, occlusion among these entities will be prevented to achieve visibility of all of them in the course of a real VS process.

To describe the camera path with boundaries on both sides, we use a path parameter $w \in [0, 1]$ 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:

$$\begin{cases} \{\mathbf{R}(0), \mathbf{t}(0)\} = \{\mathbf{I}_{3 \times 3}, \mathbf{0}_3\}, \\ \{\mathbf{R}(1), \mathbf{t}(1)\} = \{\mathbf{R}, \mathbf{t}\}. \end{cases} \quad (14)$$

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

3.2.1 Field-of-View

In arbitrary camera frame along the path, the target is expected to be in the field-of-view of the camera. For

a point \mathbf{H} mentioned in Section 2.1, its field-of-view limit in camera frame $\{\mathbf{R}(w), \mathbf{t}(w)\}$ is expressed as:

$$\begin{cases} \mathbf{e}_3^\top \mathbf{K} \mathbf{R}(w)^\top (\mathbf{H} - \mathbf{t}(w)) > 0, \\ 0 < \mathbf{e}_1^\top \mathbf{K} \mathbf{R}(w)^\top (\mathbf{H} - \mathbf{t}(w)) < \zeta_x, \\ 0 < \mathbf{e}_2^\top \mathbf{K} \mathbf{R}(w)^\top (\mathbf{H} - \mathbf{t}(w)) < \zeta_y, \end{cases} \quad (15)$$

where ζ_x, ζ_y are, respectively, the length and height of an image in pixels.

For a sphere, depth of the sphere center is meant to be larger than the sphere radius. In addition, extreme values of the projection of the sphere that is drawn in Fig. 2 are restricted to being located within the image size:

$$\begin{cases} z_o - r > 0, \\ X_{max} < \zeta_x, \\ Y_{max} < \zeta_y, \\ X_{min}, Y_{min} > 0, \end{cases} \quad (16)$$

where $X_{max}, Y_{max}, X_{min}$ and Y_{min} are computed from image moments of the sphere:

$$\begin{cases} X_{max} = \bar{X} + \sqrt{\mu_{20}} = \frac{f(x_o z_o + r \sqrt{x_o^2 + z_o^2 - r^2})}{z_o^2 - r^2} + \frac{\zeta_x}{2}, \\ X_{min} = \bar{X} - \sqrt{\mu_{20}} = \frac{f(x_o z_o - r \sqrt{x_o^2 + z_o^2 - r^2})}{z_o^2 - r^2} + \frac{\zeta_x}{2}, \\ Y_{max} = \bar{Y} + \sqrt{\mu_{02}} = \frac{f(y_o z_o + r \sqrt{y_o^2 + z_o^2 - r^2})}{z_o^2 - r^2} + \frac{\zeta_y}{2}, \\ Y_{min} = \bar{Y} - \sqrt{\mu_{02}} = \frac{f(y_o z_o - r \sqrt{y_o^2 + z_o^2 - r^2})}{z_o^2 - r^2} + \frac{\zeta_y}{2}. \end{cases} \quad (17)$$

Bring (17) into (16), we obtain a lot of inequalities in the function of the sphere center:

$$\begin{cases} \left[\frac{\zeta_x}{2f} (z_o^2 - r^2) - x_o z_o \right]^2 - r^2 (x_o^2 + z_o^2 - r^2) > 0, \\ \frac{\zeta_x}{2f} (z_o^2 - r^2) - x_o z_o > 0, \\ \left[\frac{\zeta_x}{2f} (z_o^2 - r^2) + x_o z_o \right]^2 - r^2 (x_o^2 + z_o^2 - r^2) > 0, \\ \frac{\zeta_x}{2f} (z_o^2 - r^2) + x_o z_o > 0, \\ \left[\frac{\zeta_y}{2f} (z_o^2 - r^2) - y_o z_o \right]^2 - r^2 (y_o^2 + z_o^2 - r^2) > 0, \\ \frac{\zeta_y}{2f} (z_o^2 - r^2) - y_o z_o > 0, \\ \left[\frac{\zeta_y}{2f} (z_o^2 - r^2) + y_o z_o \right]^2 - r^2 (y_o^2 + z_o^2 - r^2) > 0, \\ \frac{\zeta_y}{2f} (z_o^2 - r^2) + y_o z_o > 0. \end{cases} \quad (18)$$

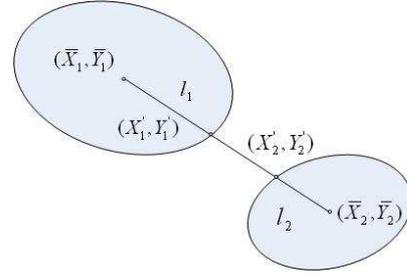


Figure 4: Projections of two spheres on the image plane.

Coordinates of the sphere center $\mathbf{o} = [x_o, y_o, z_o]^\top$ in the above function are dependent on the camera path $\{\mathbf{R}(w), \mathbf{t}(w)\}$, $w \in [0, 1]$.

3.2.2 Occlusion

Two or more entities are demanded for constituting a target in visual servoing with spheres. Occlusion among these entities is expected to be avoided to maintain the visibility of all of them. We take one sphere and a point for example, the point will not hide behind the sphere from the camera view by enforcing the following restriction in workspace:

$$(\mathbf{H} - \mathbf{o})^\top (\mathbf{t}(w) - \mathbf{o}) > 0, \quad (19)$$

where \mathbf{H} and \mathbf{o} are respectively the coordinates of a point and the sphere center expressed in the current camera frame. This inequality guarantees that the camera is above a plane that is perpendicular to the vector $(\mathbf{H} - \mathbf{o})$ and passes through the sphere center.

When two spheres are included in one target, we denote (\bar{X}_1, \bar{Y}_1) and (\bar{X}_2, \bar{Y}_2) the pixel coordinates of two centers of their projections on the image plane, as shown in Fig. 4. Line that passes through these two ellipse centers is described as:

$$Y = (X - \bar{X}_1)\alpha + \bar{Y}_1, \quad \alpha = \frac{\bar{Y}_2 - \bar{Y}_1}{\bar{X}_2 - \bar{X}_1}. \quad (20)$$

We solve equation (3) and (20) together and then calculate distance between (\bar{X}_1, \bar{Y}_1) and (\bar{X}'_1, \bar{Y}'_1) as

$$l_1 = \frac{A_1 B_1 \sqrt{\alpha^2 + 1}}{\sqrt{\mu_{02_1} + \alpha^2 \mu_{20_1} - 2\alpha \mu_{11_1}}}, \quad (21)$$

and distance between (\bar{X}_2, \bar{Y}_2) and (\bar{X}'_2, \bar{Y}'_2) as

$$l_2 = \frac{A_2 B_2 \sqrt{\alpha^2 + 1}}{\sqrt{\mu_{02_2} + \alpha^2 \mu_{20_2} - 2\alpha \mu_{11_2}}}. \quad (22)$$

As a result, occlusion between these two spheres will be avoided by imposing the following inequality:

$$\sqrt{(\bar{X}_1 - \bar{X}_2)^2 + (\bar{Y}_1 - \bar{Y}_2)^2} - l_1(\alpha) - l_2(\alpha) > 0. \quad (23)$$

Three similar inequalities will be used to avoid self-occlusion of the target when it consists of three spheres.

3.2.3 Collision

Let $\mathbf{b} \in \mathcal{R}^3$ be an obstacle in workspace expressed in the reference frame F^o . Coordinates of this obstacle in the current camera frame is $\mathbf{R}(w)^\top (\mathbf{b} - \mathbf{t}(w))$. Collision between the current camera frame and the obstacle is avoid by defining a tolerance distance d :

$$\|\mathbf{R}(w)^\top (\mathbf{b} - \mathbf{t}(w))\|^2 - d^2 > 0. \quad (24)$$

3.3 Polynomial Parametrization

We use polynomials in path parameter w to model trajectories of camera translation and rotation:

$$\begin{cases} \mathbf{q}(w) = \mathbf{U} \cdot [w^\sigma, \dots, w, 1]^\top, \\ \mathbf{t}(w) = \mathbf{V} \cdot [w^\tau, \dots, w, 1]^\top, \end{cases} \quad (25)$$

where $\mathbf{q}(w)$ is Cayley representation (Craig, 2005) of the rotation matrix:

$$[\mathbf{q}(w)]_{\times} = (\mathbf{R}(w) - \mathbf{I}_{3 \times 3})(\mathbf{R}(w) + \mathbf{I}_{3 \times 3})^{-1}. \quad (26)$$

Boundaries defined in (14) will be satisfied by assigning the first and last columns in the coefficient matrices in (25) with the following values and letting the middle of these matrices, that is $\tilde{\mathbf{U}} \in \mathcal{R}^{3 \times (\sigma-1)}$ and $\tilde{\mathbf{V}} \in \mathcal{R}^{3 \times (\tau-1)}$, to be variable:

$$\begin{cases} \mathbf{U} = [\mathbf{q}(1) - \tilde{\mathbf{U}} \cdot \mathbf{1}_{\sigma-1}, \tilde{\mathbf{U}}, \mathbf{0}_3], \\ \mathbf{V} = [\mathbf{t}(1) - \tilde{\mathbf{V}} \cdot \mathbf{1}_{\tau-1}, \tilde{\mathbf{V}}, \mathbf{0}_3]. \end{cases} \quad (27)$$

Matrices $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ will be assigned a "guess" value and substituted in (25) and then (16), (23) and (19) to calculate the left hand side of all the inequalities for all the entities and all the way along the path with $w \in (0, 1)$. A minimum value of the left hand side of all the inequalities will be obtained. If this minimum value is negative, nonlinear optimization will be used to search for appropriate values for $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ until the minimum of all the calculation is positive. Till this end, all the constraints will be satisfied.

Planned image trajectories will be generated by bringing new values of $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ into (25), (3) and (4). These trajectories will be followed by an IBVS controller (Mezouar and Chaumette, 2002a):

$$\mathbf{T}_c = -\lambda_1 \hat{\mathbf{L}}^+ (\mathbf{s}(t) - \mathbf{s}_p(w)) + \hat{\mathbf{L}}^+ \frac{d\mathbf{s}_p(w)}{dt}, \quad (28)$$

where the feature set is determined as specified in Section 3.1 with $\mathbf{s}(t)$ holding the current values and $\mathbf{s}_p(w)$ being the planned ones when $w = 1 - e^{-t\lambda_2}$. Interaction matrix or image jacobian (Mezouar and Chaumette, 2002b) regarding the features of a sphere

is a stack of matrices $\mathbf{L}_{s_{12}}$ and \mathbf{L}_{s_3} . Matrix $\mathbf{L}_{s_{12}}$ corresponds to the centroid of the projection of the sphere:

$$\mathbf{L}_{s_{12}} = \begin{bmatrix} -1/\bar{z} & 0 \\ 0 & -1/\bar{z} \\ \bar{X}/\bar{z} + a\mu_{20} + b\mu_{11} & \bar{Y}/\bar{z} + a\mu_{11} + b\mu_{02} \\ \bar{X}\bar{Y} + \mu_{11} & 1 + \bar{Y}^2 + \mu_{02} \\ -1 - \bar{X}^2 - \mu_{20} & -\bar{X}\bar{Y} - \mu_{11} \\ \bar{Y} & -\bar{X} \end{bmatrix}^\top,$$

where \bar{z} is not the depth of the sphere center, and

$$\begin{cases} \bar{z} = \frac{1}{a\bar{X} + b\bar{Y} + c} = z_o - r^2/z_o, \\ a = x_o/(x_o^2 + y_o^2 + z_o^2 - r^2), \\ b = y_o/(x_o^2 + y_o^2 + z_o^2 - r^2), \\ c = z_o/(x_o^2 + y_o^2 + z_o^2 - r^2), \end{cases} \quad (29)$$

while \mathbf{L}_{s_3} corresponds to the half of the sum of μ_{02} and μ_{20} :

$$\mathbf{L}_{s_3} = \begin{bmatrix} -a\mu_{20} - b\mu_{11} \\ -a\mu_{11} - b\mu_{02} \\ (\frac{1}{\bar{z}} + a\bar{X})\mu_{20} + (\frac{1}{\bar{z}} + b\bar{Y})\mu_{02} + \mu_{11}(b\bar{X} + a\bar{Y}) \\ 2\bar{Y}\mu_{02} + \bar{X}\mu_{11} + \bar{Y}\mu_{20} \\ -2\bar{X}\mu_{20} - \bar{Y}\mu_{11} - \bar{X}\mu_{02} \\ 0 \end{bmatrix}^\top.$$

The second addend in (28) is a compensation item provided by the derivation of $\mathbf{e} = \hat{\mathbf{L}}^+ (\mathbf{s}(t) - \mathbf{s}_p(w))$ (Mezouar and Chaumette, 2002a), and in this paper we have:

$$\frac{d\mathbf{s}_p(w)}{dt} = \frac{d\mathbf{s}_p(w)}{dw} \cdot \frac{dw}{dt} = \lambda_2 e^{-t\lambda_2} \frac{d\mathbf{s}_p(w)}{dw}.$$

For features selected for a sphere, their derivatives with respect to the path parameter w are listed:

$$\begin{aligned} \frac{d\bar{X}}{dw} &= \frac{f}{(z_o^2 - r^2)^2} \left[z_o(z_o^2 - r^2) \frac{dx_o}{dw} - x_o(z_o^2 - r^2) \frac{dz_o}{dw} \right], \\ \frac{d\bar{Y}}{dw} &= \frac{f}{(z_o^2 - r^2)^2} \left[z_o(z_o^2 - r^2) \frac{dy_o}{dw} - y_o(z_o^2 - r^2) \frac{dz_o}{dw} \right], \\ \frac{d\mu_{02}}{dw} &= \frac{2f^2 r^2}{(z_o^2 - r^2)^3} \left[y_o(z_o^2 - r^2) \frac{dy_o}{dw} - z_o(2y_o^2 + z_o^2 - r^2) \frac{dz_o}{dw} \right], \\ \frac{d\mu_{20}}{dw} &= \frac{2f^2 r^2}{(z_o^2 - r^2)^3} \left[x_o(z_o^2 - r^2) \frac{dx_o}{dw} - z_o(2x_o^2 + z_o^2 - r^2) \frac{dz_o}{dw} \right], \\ \frac{d}{dw} \left(\frac{\mu_{20} + \mu_{02}}{2} \right) &= \frac{1}{2} \left(\frac{d\mu_{02}}{dw} + \frac{d\mu_{20}}{dw} \right). \end{aligned}$$

4 EXAMPLES

Synthetic scene is generated using MATLAB to verify the proposed scheme. The first example aims to avoid collision in the Cartesian space while keeping visible all the entities that consist of a target. The second example gives a planned path that prevents occlusion

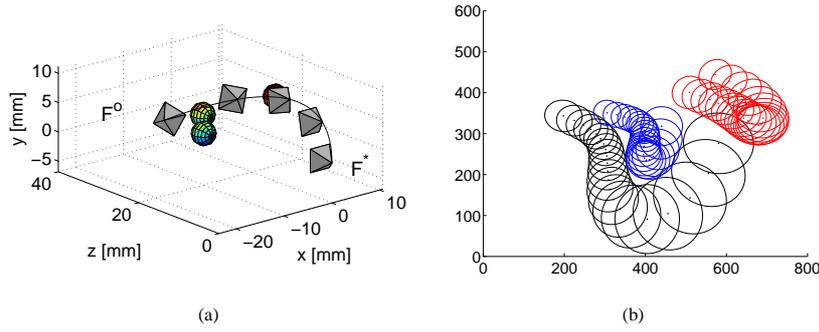


Figure 5: Occlusion avoidance among three spheres. (a) Camera path. (b) Image trajectories.

among three spheres and simultaneously meets field-of-view limit for all of them. Whatever the selected features are in an image of size 800×600 pixels, normally distributed image noises range from $(-1, 1)$ pixels are added upon them.

4.1 Collision Avoidance

In this example, the target consist of a sphere and two points. As depicted in Fig. 6 (a), the sphere has radius of 3 mm and its center coordinates as $[0, 0, 20]^T$ mm expressed in F^* . The other two points are symbolized respectively as a star and an x-mark beside the sphere. An obstacle represented by a box is found between the two camera poses $F^* = \{\mathbf{I}_{3 \times 3}, \mathbf{0}_3\}$ and $F^o = \{e^{[\rho] \times}, [-35, 10, 10]^T \text{ mm}\}$ with $\rho = [\pi/12, \pi/4, -\pi/6]$. Intrinsic parameters of the camera are estimated as follows:

$$\mathbf{K} = \begin{pmatrix} 508 & 0 & 403 \\ 0 & 496 & 302 \\ 0 & 0 & 1 \end{pmatrix}. \quad (30)$$

Camera views of the target obtained in F^o and F^* are separately displayed in Fig. 6 (b) and (c). Features extracted from these camera views are pixel coordinates of the centroid of the ellipse, that is \bar{X} and \bar{Y} , combination of the length of the major and minor semi-axes of the ellipse, that is $(A^2 + B^2)/2$, and pixel coordinates of image projections of the other two points. This is exactly the same as the feature set mentioned in (8). Given the position and model of the target, relative camera pose between F^* and F^o is estimated using a virtual VS method and is used as boundaries for the planned camera path. Fig. 6 (d) plots the satisfactory camera path that avoids collision with a given obstacle and keeps the sphere and also the other two points in the field-of-view of the camera. It is also noticed that the two other points are not occluded by the sphere along the camera path by applying constraints as shown in (19). Fig. 6 (e) demonstrates visibility of all the entities that consist

of the target in this example, where circles are used to imply the start of image trajectories of the points and diamonds the end of them.

4.2 Occlusion Avoidance

The scenario for visual servoing path-planning with three spheres is illustrated in Fig. 3 (a), where these three spheres have the same radius of 2 mm and positions of their centers as $[-10, 2, 20]^T$, $[-6, 3, 25]^T$ and $[5, 5, 20]^T$ expressed in F^* . Occlusion avoidance among these spheres and the maintenance of camera view of them are expected to be achieved during a real time VS. Boundaries of this camera path are demonstrated in Fig. 3 (a) as $F^* = \{\mathbf{I}_{3 \times 3}, \mathbf{0}_3\}$ and $F^o = \{e^{[\rho] \times}, [-20, 7, 18]^T \text{ mm}\}$ with $\rho = [0, \pi/3, -\pi/6]$. Intrinsic parameters of the camera are approximated as:

$$\mathbf{K} = \begin{pmatrix} 410 & 0 & 405 \\ 0 & 380 & 305 \\ 0 & 0 & 1 \end{pmatrix}.$$

Relative camera pose between F^* and F^o is estimated using a virtual VS method based on the feature set stated in (9). This feature set is a stack of \bar{X} , \bar{Y} and $(A^2 + B^2)/2$ for each sphere and can be extracted from the projection of the sphere in F^o and F^* as shown in Fig. 3 (b)-(c). The corresponding virtual camera path and image trajectories are also displayed separately in Fig. 3 (d) and (e). It is obvious that in Fig. 3 (e) there are two spheres occluded with each other, the case of which will fail a real VS task. Therefore, the proposed path-planning approach is intended to generate another camera path meeting the required constraints in this example. The resulting camera path and image trajectories are shown in Fig. 5 (a)-(b), where occlusions are perfectly avoided with trajectories of all the spheres kept in the field-of-view of the camera.

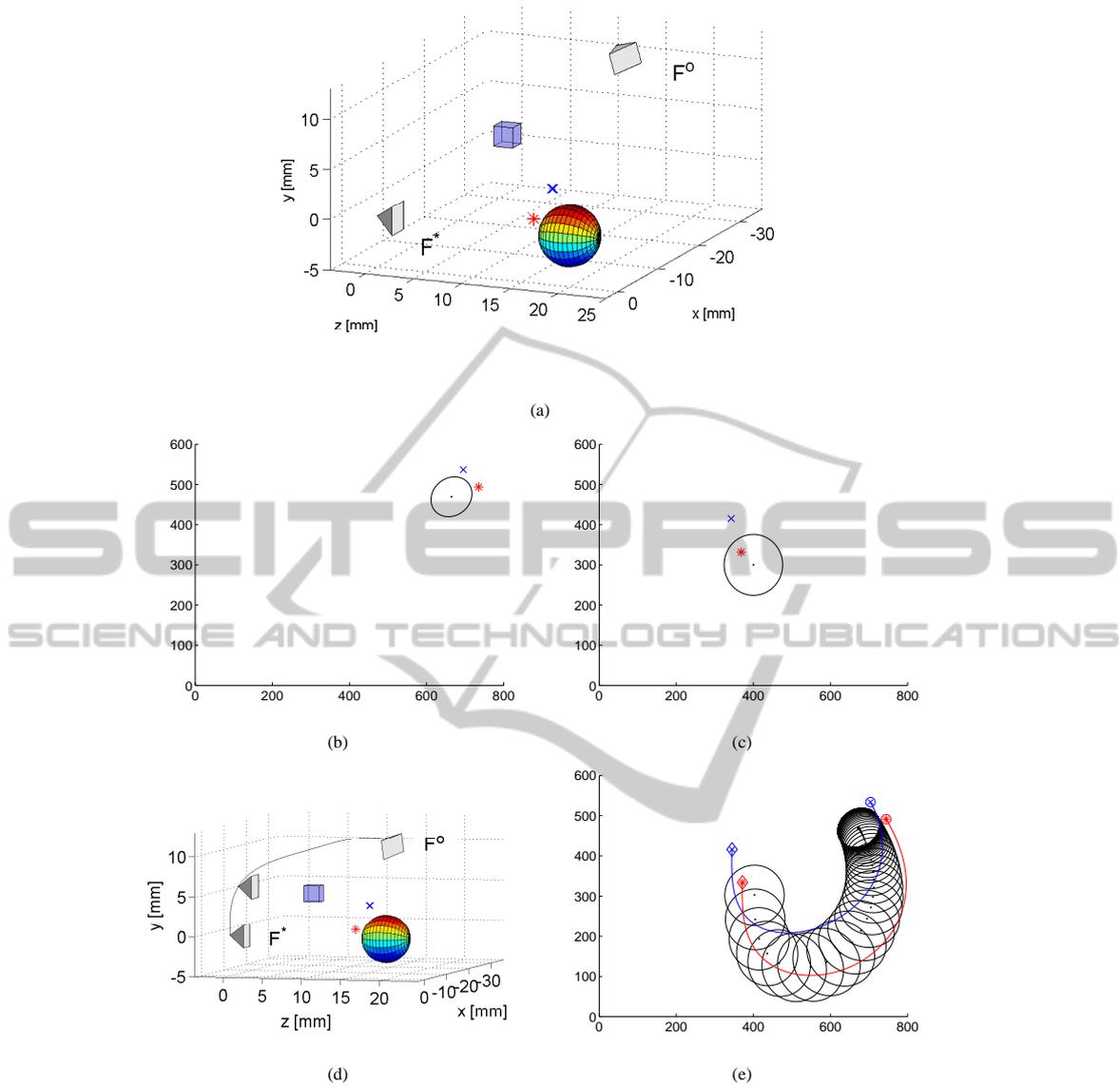


Figure 6: Collision avoidance in workspace. (a) Scenario (b) Camera view in F^o . (c) Camera view in F^* . (d) Camera path. (e) Image trajectories.

5 CONCLUSIONS

This paper proposes a path-planning approach for visual servoing with spheres. Sphere is a very special object that projects in the image plane as an ellipse whose centroid may not correspond to the center of the sphere. One sphere only provides three independent features. Therefore, this paper combines a sphere with at least two points or at least two other spheres to serve as a target. In each of the two cases, camera view of the whole target is obtained and also the sphere is prevented occluding other entities in the target. Simulations validate the proposed path-planning

approach. Future work is meant to include constraints in joint space. Also, path-planning from objects that are more complicated than a sphere, such as a bottle, are interesting to explore.

REFERENCES

- Chaumette, F. (1998a). *De la perception l'action : l'asservissement visuel; de l'action la perception : la vision active*. PhD thesis, Universit 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, S., editors, *The Confluence of Vision and Control*, pages 66–78. vol. 237 of Lecture Notes in Control and Information Sciences, Springer-Verlag.
- Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. *IEEE Trans. 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. (2009). Visual servoing path-planning via homogeneous forms and lmi optimizations. *IEEE Trans. on Robotics*, 25(2):281–291.
- Chesi, G. and Hashimoto, K., editors (2010). *Visual Servoing via Advanced Numerical Methods*. Springer.
- 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 Trans. on Robotics*, 20(5):908–914.
- Chesi, G. and Hung, Y. S. (2007). Global path-planning for constrained and optimal visual servoing. *IEEE Trans. on Robotics*, 23(5):1050–1060.
- Chesi, G. and Vicino, A. (2004). Visual servoing for large camera displacements. *IEEE Trans. on Robotics*, 20(4):724–735.
- Collewet, C. and Marchand, E. (2010). Luminance: A new visual feature for visual servoing. In Chesi, G. and Hashimoto, K., editors, *Visual Servoing via Advanced Numerical Methods*, pages 71–90. vol. 401 of Lecture Notes in Control and Information Sciences, Springer-Verlag.
- Cowan, N., Weingarten, J., and Koditschek, D. (2002). Visual servoing via navigation functions. *IEEE Trans. on Robotics and Automation*, 18(4):521–533.
- Craig, J., editor (2005). *Introduction to Robotics: Mechanics and Control*. Pearson Education.
- Gans, N. and Hutchinson, S. (2007). Stable visual servoing through hybrid switched-system control. *IEEE Trans. on Robotics*, 23(3):530–540.
- Hashimoto, K., Kimoto, T., Ebine, T., and Kimura, H. (1991). Manipulator control with image-based visual servo. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2267–2272, San Francisco, CA.
- Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250.
- Mezouar, Y. and Chaumette, F. (2002a). Path planning for robust image-based control. *IEEE Trans. on Robotics and Automation*, 18(4):534–549.
- Mezouar, Y. and Chaumette, F. (2002b). Path planning for robust image-based control. *IEEE Trans. on Robotics and Automation*, 18(4):534–549.
- Oh, P. Y. and Allen, P. K. (2001). Visual servoing by partitioning degrees of freedom. *IEEE Trans. on Robotics and Automation*, 17(1):1–17.
- Shen, T. and Chesi, G. (2012). Visual servoing path-planning for cameras obeying the unified model. *Advanced Robotics*, 26(8–9):843–860.
- Tahri, O. and Chaumette, F. (2005). Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. 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 Hashimoto, K., editors, *Visual Servoing via Advanced Numerical Methods*, pages 231–252. vol. 401 of Lecture Notes in Control and Information Sciences, Springer-Verlag.
- Tatsambon Fomena, R. and Chaumette, F. (2008). Visual servoing from two special compounds of features using a spherical projection model. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'08*, pages 3040–3045, Nice, France.
- Taylor, C. and Ostrowski, J. (2000). Robust vision-based pose control. In *IEEE Int. Conf. Robot. Automat.*, pages 2734–2740, San Francisco, CA.