2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Visual servoing: a global path-planning approach

G. Chesi, Y.S. Hung

Abstract— **This paper considers the problem of realizing visual servoing taking into account constraints such as visibility and workspace constraints while minimizing a cost function such as spanned image area and trajectory length. A new path-planning scheme is proposed by, first, introducing a robust object reconstruction which allows one to obtain feasible image trajectories. Second, the rotation path is parameterized through a particular extension of the Euler parameters in order to obtain an equivalent expression of the rotation matrix as a quadratic function of unconstrained variables, hence largely simplified with respect to standard parameterizations which involve transcendental functions. Then, polynomials of arbitrary degree are used to complete the parametrization and formulate a general optimization where a number of constraints and costs can be considered. The optimal trajectory is followed by tracking the image trajectories with standard IBVS controllers.**

Keywords: Visual servoing, Teaching-by-showing approach, Constraints, Costs, Path-planning.

I. INTRODUCTION

In recent years, the "teaching-by-showing" approach has received an increasing attention. It consists of teaching the desired location for an eye-in-hand robotic system by showing the view of some reference features in such a location. The camera is then moved to another location from which must be steered to the desired location by exploiting the current and desired view of the features. See for example [5], [7], [10] for detailed classifications of visual servoing approaches. Several methods to deal with this task have been proposed. Some examples are the position-based visual servoing (PBVS) where the feedback error is the camera pose (see for example [18], [19]), and the image-based visual servoing (IBVS) where the feedback error is the image error (see for example [6], [16]). In the 2 $1/2$ D visual servoing [10] the feedback error contains both the camera pose and the image error. Then, other methods exploit partitioning techniques [2], [14], navigation functions [3], invariance with respect to the intrinsic parameters [8], image moments [17], generation of circular-like trajectories [1].

When controlling a robot, convergence and robustness are not the only important issues. In fact, workspace limits should be taken into account in order to obtain a trajectory that does not make the robot end-effector colliding with obstacles present in the scene. Analogously, joint limits should be considered because the robot could be unable to reach a certain location due to its particular structure. And, besides issues concerning the physical constraints, there are also issues concerning the performance. In fact, when more than one feasible robot trajectory exist, one could also be interested in finding the shortest trajectory, or the smoothest trajectory, or the trajectory which maximizes the visibility margin.

This paper proposes a new path-planning technique in the image space which allows one to consider constraints such as visibility and workspace constraints, together with the objective of minimizing trajectory costs such as length and spanned image area. This is achieved by parameterizing in the six-dimensional rigid motion space all the trajectories connecting the initial to the desired location which, although unknown, can be relatively computed through an object reconstruction from image measurements and, if available, also a cad model of the observed object. In order to deal with calibration errors and image noise, a new robust object reconstruction is proposed which allows one to obtain image trajectories satisfying the boundary conditions. In order to obtain functions that can be efficiently handled in optimization tools, the rotation path is parameterized through a particular extension of the Euler parameters which allows one to obtain an equivalent expression of the rotation as a quadratic function of unconstrained variables. Polynomials of arbitrary degree are then used to complete the parametrization. Once the image trajectories have been computed, the camera is steered to the desired location by using standard IBVS controllers.

It is worthwhile to notice that other path-planning methods have been proposed in [11], [12], [15], [20] which solve related problems by exploiting, in a discretized framework, optimal control formulations, screw motions, and geodesic paths modulated by repulsive potential fields respectively. The approach in this paper proposes a new strategy which consists of parameterizing all the possible trajectories and makes possible considering a number of constraints to fulfill and costs to minimize. This is achieved by using a new approach based on parameter-dependent object reconstruction and extended Euler parameters.

II. PRELIMINARIES

Let us introduce the following notation. Let $\mathbb{R}$ denote the real number set, $SO(3)$ the set of rotation matrices in $\mathbb{R}^{3\times 3}$, and $SE(3)$ the cartesian product $SO(3) \times \mathbb{R}^3$. We denote with $\mathbf{I}_n$ the identity matrix $n \times n$, $\mathbf{0}_n$ the null vector $n \times 1$, $\mathbf{1}_n$ the vector $n \times 1$ with all elements equal to 1, $\mathbf{e}_i$ the i-th column of $\mathbf{I}_3$, and $[\mathbf{v}]_\times$ the skew-symmetric matrix of $\mathbf{v} \in \mathbb{R}^3$. Moreover:

- $\mathbf{A} \in \mathbb{R}^{3\times 3}$: upper triangular intrinsic parameters matrix;

G. Chesi (corresponding author) and Y.S. Hung are with the Department of Electrical and Electronic Engineering, University of Hong Kong, E-mail: {chesi,yshung}@eee.hku.hk

- $\mathcal{F}^\circ, \mathcal{F}^*$: initial and desired camera frames with pose $\{\mathbf{O}^\circ, \mathbf{c}^\circ\}, \{\mathbf{O}^*, \mathbf{c}^*\} \in SE(3)$ with respect to the absolute frame;
- $\boldsymbol{\rho}_i \in \mathbb{R}^3$: i-th point in the three-dimensional space expressed with respect to the absolute frame;
- $\mathbf{p}_i^\circ = [x_i^\circ, y_i^\circ, 1]', \mathbf{p}_i^* = [x_i^*, y_i^*, 1]' \in \mathbb{R}^3$: projections in pixel coordinates of the i-th point on $\mathcal{F}^\circ$ and $\mathcal{F}^*$ according to

$$\begin{cases} \alpha_i^\circ \mathbf{p}_i^\circ &= \mathbf{A}\mathbf{O}^{\circ T}(\boldsymbol{\rho}_i - \mathbf{c}^\circ) \\ \alpha_i^* \mathbf{p}_i^* &= \mathbf{A}\mathbf{O}^{*T}(\boldsymbol{\rho}_i - \mathbf{c}^*) \end{cases} \tag{1}$$

  where $\alpha_i^\circ, \alpha_i^* \in \mathbb{R}$ are the point depths;
- $\mathcal{P}(\mathcal{F}^*, \mathcal{F}^\circ) \in SE(3)$: camera pose $\{\mathbf{R}, \mathbf{t}\}$ of $\mathcal{F}^*$ with respect to $\mathcal{F}^\circ$, according to

$$\begin{cases} \mathbf{R} &= \mathbf{O}^{\circ T}\mathbf{O}^* \\ \mathbf{t} &= \mathbf{O}^{\circ T}(\mathbf{c}^* - \mathbf{c}^\circ) \end{cases} \tag{2}$$

Let us suppose that a set $\mathcal{S} = \{(\mathbf{p}_i^\circ, \mathbf{p}_i^*),\ i = 1, \ldots, n\}$ of n object point correspondences is available. The problem consists of steering the camera from the initial to the desired location satisfying constraints such as visibility and workspace constraints while optimizing a certain trajectory cost.

III. PATH-PLANNING

The strategy proposed in this paper consists of generating trajectories of the object points in the image and then tracking them by using IBVS controllers. We indicate these image trajectories with $\mathbf{p}_i(w)$ where $w \in [0,1]$ is the trajectory abscise, with $w = 0$ indicating the initial location and $w = 1$ the desired location. The vector $\mathbf{p}_i(w)$ must satisfy the boundary conditions

$$\begin{cases} \mathbf{p}_i(0) &= \mathbf{p}_i^\circ \\ \mathbf{p}_i(1) &= \mathbf{p}_i^* \end{cases} \tag{3}$$

The above conditions are not the only constraints that $\mathbf{p}_i(w)$ must satisfy. In fact, the set of $\mathbf{p}_i(w)$, $i = 1, \ldots, n$, must be such that there exists a parameter-dependent camera frame from which the observed object points match the $\mathbf{p}_i(w)$ for all $w \in [0,1]$. In order to cope with this problem as well as facilitate the task of taking into account constraints and trajectory costs defined outside the image space, we introduce a new parametrization as described in the following sections.

A. Trajectory parametrization

From $\mathcal{S}$, $\mathbf{A}$, and the cad model of the object (that is the set of physical points r_i), one can calculate the camera pose $\{\mathbf{R}, \mathbf{t}\}$ by solving (1)–(2) through linear least-squares techniques. If the cad model of the object is not available, $\mathbf{t}$ can be computed only up to a scale factor which stands for the unknown distance between the initial and desired frame origins. Indeed, the normalized camera pose $\{\mathbf{R}, \mathbf{t}_{norm}\}$ with $\mathbf{t}_{norm} = \mathbf{t}/\|\mathbf{t}\|$ can be computed through the essential matrix algorithm or the homography matrix algorithm relative to a virtual plane in the case of non coplanar features supposing $n \geq 8$. If the features are known to be coplanar, the camera pose can be computed through the homography matrix algorithm supposing $n \geq 4$. See [4] and [9] for details. For pure rotation motion, i.e. $\mathbf{t} = \mathbf{0}_3$, the normalized translation is defined as $\mathbf{t}_{norm} = \mathbf{0}_3$.

Let $\{\mathbf{R}, \mathbf{d}\}$ be the estimated camera pose, being $\mathbf{d}$ either the physical translation $\mathbf{t}$ or the normalized translation $\mathbf{t}_{norm}$. Let $\mathcal{F}_d(w)$ be the camera frame in the *reconstruction space* (that is relative to the available translation $\mathbf{d}$). Let us define the pose of $\mathcal{F}_d(w)$ with respect to $\mathcal{F}^\circ$ as

$$\mathcal{P}(\mathcal{F}_d(w), \mathcal{F}^\circ) = \{\mathbf{R}(w), \mathbf{d}(w)\}. \tag{4}$$

At the extreme points of the trajectory this pose must satisfy

$$\begin{cases} \{\mathbf{R}(0), \mathbf{d}(0)\} &= \{\mathbf{I}_3, \mathbf{0}_3\} \\ \{\mathbf{R}(1), \mathbf{d}(1)\} &= \{\mathbf{R}, \mathbf{d}\} \end{cases} \tag{5}$$

In ideal conditions, that is in the absence of calibration errors and image noise, the trajectory of the i-th object point can be expressed in function of $\mathbf{R}(w)$ and $\mathbf{d}(w)$ as

$$\bar{\alpha}_i(w)\mathbf{p}_i(w) = \mathbf{A}\mathbf{R}(w)^T(\mathbf{u}_i - \mathbf{d}(w)). \tag{6}$$

where $\bar{\alpha}_i(w)$ is the parameterized point depth and $\mathbf{u}_i$ is the i-th physical point $\boldsymbol{\rho}_i$ in the reconstruction space known as *object reconstruction*. This object reconstruction can be computed by solving the system

$$\begin{cases} \bar{\alpha}_i(0)\mathbf{p}_i^\circ &= \mathbf{A}\mathbf{u}_i \\ \bar{\alpha}_i(1)\mathbf{p}_i^* &= \mathbf{A}\mathbf{R}^T(\mathbf{u}_i - \mathbf{d}) \end{cases} \tag{7}$$

which amounts to calculating a SVD by eliminating the point depths $\bar{\alpha}_i(0), \bar{\alpha}_i(1)$. Hence, the sought trajectory can be expressed as the solution of the following constrained minimization problem:

$$\min_{\mathbf{R}(w):\mathbb{R}\to SO(3),\mathbf{d}(w):\mathbb{R}\to\mathbb{R}^3} h(\mathbf{R}(w), \mathbf{d}(w))$$
$$\text{s.t.} \begin{cases} \text{pose boundary conditions (5)} \\ \text{image boundary conditions (3)} \\ g_k(w, \mathbf{R}(w), \mathbf{d}(w)) > 0\ \forall w \in [0,1]\ \forall k = 1, \ldots, m_g \end{cases} \tag{8}$$

where $g_k(w)$, $k = 1, \ldots, m_g$, indicate the constraints that the camera must satisfy along the trajectory, and h is the cost function to be minimized by the trajectory. In order to establish if $g_k(w)$ is positive for all $w \in [0,1]$, one must evaluate, at each step of the optimization procedure, each function $g_k(w)$ at $w = 0$, at $w = 1$, and at the points in $[0,1]$ where its derivative vanishes in order to find all the global minima. However, this turns out to be difficult and computationally heavy with standard parameterizations because $\mathbf{R}(w)$, and consequently $g_k(w)$, present transcendental terms (see for example exponential coordinates, x-y-z angles, etc...). In order to cope with this problem, we will derive in the sequel a special parametrization of the trajectories for which the functions $g_k(w)$ are polynomial in w.

B. Robust object reconstruction

In the presence of calibration errors and image noise, the system (7) may admit no solution for $\mathbf{u}_i$. Clearly, a least-squares solution can still be computed through SVD, but this solution cannot allow the image trajectory $\mathbf{p}_i(w)$ provided by (6) to satisfy the image boundary conditions (3).

Therefore, we look for a *robust object reconstruction* which minimizes the effect of uncertainties on our robot control. In particular, we introduce a parameter-dependent object reconstruction $\mathbf{u}_i(w)$ satisfying

$$\left\{ \begin{array}{rcl} \tilde{\alpha}_i(0)\mathbf{p}_i^\circ & = & \mathbf{A}\mathbf{u}_i(0) \\ \tilde{\alpha}_i(1)\mathbf{p}_i^* & = & \mathbf{A}\mathbf{R}^T\left(\mathbf{u}_i(1) - \mathbf{d}\right) \end{array} \right. \tag{9}$$

for some point depths $\tilde{\alpha}_i(0), \tilde{\alpha}_i(1)$. Among all possible solutions $\mathbf{u}_i(w)$ for the above system, we select the linear solution because it is the simplest and because it is the closest to the ideal constant $\mathbf{u}_i$ in (7). Hence, let us write $\mathbf{u}_i(w)$ as

$$\mathbf{u}_i(w) = (1-w)\mathbf{a}_i + \mathbf{b}_i. \tag{10}$$

Since we are interested in the solution closest to the constant one, we aim to find the $\mathbf{u}_i(w)$ with the smallest $\mathbf{a}_i$ which satisfies (9), that is

$$\min_{\mathbf{a}_i, \mathbf{b}_i} \|\mathbf{a}_i\| \text{ s.t. (9)–(10).} \tag{11}$$

The above optimization problem can be easily solve in closed-form through Lagrange multipliers. The image projection $\mathbf{p}_i(w)$ along the trajectory is then given by

$$\tilde{\alpha}_i(w)\mathbf{p}_i(w) = \mathbf{A}\mathbf{R}(w)^T\left(\mathbf{u}_i(w) - \mathbf{d}(w)\right). \tag{12}$$

C. Rotation parameter

We start by considering the representation of rotation matrices through the Euler parameters (see for example [13]). According to this representation, any rotation matrix can be represented as

$$\mathbf{\Lambda}(\boldsymbol{\phi}) = \begin{bmatrix} \phi_1^2 - \phi_2^2 - \phi_3^2 + \phi_4^2 & 2(\phi_1\phi_2 - \phi_3\phi_4) & 2(\phi_1\phi_3 + \phi_2\phi_4) \\ 2(\phi_1\phi_2 + \phi_3\phi_4) & -\phi_1^2 + \phi_2^2 - \phi_3^2 + \phi_4^2 & 2(\phi_2\phi_3 - \phi_1\phi_4) \\ 2(\phi_1\phi_3 - \phi_2\phi_4) & 2(\phi_2\phi_3 + \phi_1\phi_4) & -\phi_1^2 - \phi_2^2 + \phi_3^2 + \phi_4^2 \end{bmatrix} \tag{13}$$

for some $\boldsymbol{\phi} \in \mathbb{R}^4$ satisfying $\|\boldsymbol{\phi}\| = 1$. Also, $\mathbf{\Lambda}(\boldsymbol{\phi})$ is a rotation matrix for any $\boldsymbol{\phi}$ satisfying $\|\boldsymbol{\phi}\| = 1$. The Euler parameters corresponding to $\mathbf{R}$ can be found as $\boldsymbol{\phi} = \xi(\mathbf{R})$ where

$$\boldsymbol{\xi}(\mathbf{R}) = \left[\sin\frac{\theta}{2}\,\mathbf{u}^T, \cos\frac{\theta}{2}\right]^T \tag{14}$$

where $\theta \in [0, \pi]$ and $\mathbf{u} \in \mathbb{R}^3$, $\|\mathbf{u}\| = 1$, are respectively the rotation angle and axis in the exponential coordinates of $\mathbf{R}$, i.e. $\mathbf{R} = e^{[\boldsymbol{\theta}\mathbf{u}]_\times}$.

Hence, one could parameterize the rotation through (13) and the vector $\boldsymbol{\phi}$. However, $\boldsymbol{\phi}$ is not free due to the constraint $\|\boldsymbol{\phi}\| = 1$. Clearly, one could then parameterize $\boldsymbol{\phi}$ in order to ensure this constraint, but this would require the introduction of irrational or trigonometrical functions.

Therefore, in order to get rid of the constraint $\|\boldsymbol{\phi}\| = 1$ and obtain a rational parametrization of the rotation, we introduce the following extended Euler parametrization:

$$\mathbf{\Phi}(\boldsymbol{\phi}) = \frac{1}{\|\boldsymbol{\phi}\|^2}\mathbf{\Lambda}(\boldsymbol{\phi}). \tag{15}$$

This parametrization satisfies the following properties. First, for any rotation matrix $\mathbf{R}$ there exists $\boldsymbol{\phi}$ such that $\mathbf{R} = \mathbf{\Phi}(\boldsymbol{\phi})$. In particular, $\mathbf{R} = \mathbf{\Phi}(\gamma\boldsymbol{\xi}(\mathbf{R}))\ \forall\gamma \neq 0$. Second, $\mathbf{\Phi}(\boldsymbol{\phi})$ is a rotation matrix for all $\boldsymbol{\phi} \neq \mathbf{0}_4$.

We observe that we have got rid of the constraint $\|\boldsymbol{\phi}\| = 1$ at the expense of the denominator in (15). As it will become clear in the next sections, this denominator does not affect the optimization problem. We also observe that we have derived a parametrization of the rotation based on four free parameters while there exist parameterizations based on three parameters only such that the exponential coordinates θ and $\mathbf{u}$ previously mentioned. These parameterizations can be equivalently used in the problem we are going to formulate through suitable variables transformations which allow one to finally derive a rational expression for the rotation, however this will result in higher degree polynomials also affected by some degenerate configurations which require a separate investigation.

D. Polynomial parametrization

Let us parameterize the rotation of the camera frame $\mathcal{F}_d(w)$ as

$$\mathbf{R}(w) = \mathbf{\Phi}(\boldsymbol{\phi}(w)) \tag{16}$$

where $\boldsymbol{\phi}(w)$ is a parameterized extended Euler parameter. The camera pose along the trajectory is hence described by $\boldsymbol{\phi}(w)$ and $\mathbf{d}(w)$. Let us express these vectors as polynomials according to

$$\left\{ \begin{array}{rcl} \boldsymbol{\phi}(w) & = & \tilde{\mathbf{M}}\left[w^{\delta_M}, w^{\delta_M - 1}, \ldots, w, 1\right]^T \\ d(w) & = & \tilde{\mathbf{N}}\left[w^{\delta_N}, w^{\delta_N - 1}, \ldots, w, 1\right]^T \end{array} \right. \tag{17}$$

where $\tilde{\mathbf{M}} \in \mathbb{R}^{4 \times \delta_M + 1}$ and $\tilde{\mathbf{N}} \in \mathbb{R}^{3 \times \delta_N + 1}$ are coefficient matrices. In order to satisfy the rotation boundary conditions in (5), we impose

$$\left\{ \begin{array}{rcl} \boldsymbol{\phi}(0) & = & [0, 0, 0, 1]^T \\ \boldsymbol{\phi}(1) & = & \boldsymbol{\xi}(\mathbf{R}) \end{array} \right. \tag{18}$$

(clearly, one can equivalently impose the same quantity scaled by the same factor γ with $\gamma \neq 0$). Then, taking into account (17), one has that the pose boundary conditions (5) are satisfied if and only if

$$\left\{ \begin{array}{rclrcl} \tilde{\mathbf{M}}\left[\mathbf{0}_{\delta_M}^T, 1\right]^T & = & [\mathbf{0}_3^T, 1]^T, & \tilde{\mathbf{N}}\left[\mathbf{0}_{\delta_N}^T, 1\right]^T & = & \mathbf{0}_3 \\ \tilde{\mathbf{M}}\mathbf{1}_{\delta_M + 1} & = & \boldsymbol{\xi}(\mathbf{R}), & \tilde{\mathbf{N}}\mathbf{1}_{\delta_N + 1} & = & \mathbf{d} \end{array} \right. \tag{19}$$

which imply that the matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{N}}$ can be parameterized as

$$\left\{ \begin{array}{rcl} \tilde{\mathbf{M}} & = & [\boldsymbol{\xi}(\mathbf{R}) - \mathbf{M}\mathbf{1}_{\delta_M - 1} - [\mathbf{0}_3^T, 1]^T, \mathbf{M}, [\mathbf{0}_3^T, 1]^T] \\ \tilde{\mathbf{N}} & = & [\mathbf{d} - \mathbf{N}\mathbf{1}_{\delta_N - 1}, \mathbf{N}, \mathbf{0}_3] \end{array} \right. \tag{20}$$

where $\mathbf{M} \in \mathbb{R}^{4 \times \delta_M - 1}$ and $\mathbf{N} \in \mathbb{R}^{3 \times \delta_N - 1}$ are free matricial parameters. Therefore, the camera pose along the trajectory is parameterized by the matrices $\mathbf{M}$ and $\mathbf{N}$, and the optimization problem (8) can be rewritten as

$$\begin{array}{c} \displaystyle\min_{\mathbf{M} \in \mathbb{R}^{4 \times \delta_M - 1}, \mathbf{N} \in \mathbb{R}^{3 \times \delta_N - 1}} h(\mathbf{M}, \mathbf{N}) \\ \text{s.t. } g_k(w, \mathbf{M}, \mathbf{N}) > 0\ \forall w \in [0, 1]\ \forall k = 1, \ldots, n_g \end{array} \tag{21}$$

Let us observe that the boundary conditions (5) and (3) are implicitly satisfied in (21).

IV. Constraints and costs

A. Constraints

It is possible to include several constraints in (21) depending on the specific problem. Here we describe visibility and workspace constraints. Similarly one can include also joint constraints.

Visibility constraints. Let us write the image projection $\mathbf{p}_i(w)$ as $\mathbf{p}_i(w) = [x_i(w), y_i(w), 1]$. In order to guarantee that the image projections along the trajectory remain in the field of view, we have to introduce the visibility constraint

$$\left.\begin{array}{c} x_{min} < x_i(w) < x_{max} \\ y_{min} < y_i(w) < y_{max} \end{array}\right\} \; \forall w \in [0,1] \; \forall i = 1,\ldots,n \tag{22}$$

where $x_{min}, x_{max}, y_{min}, y_{max} \in \mathbb{R}$ are the screen limits. By eliminating the point depth $\tilde{\alpha}_i(w)$ in (12) one obtains

$$\left\{\begin{array}{rcl} x_i(w) & = & \dfrac{\mathbf{e}_1^T \mathbf{A}\mathbf{R}(w)^T\left(\mathbf{u}_i(w) - \mathbf{d}(w)\right)}{\mathbf{e}_3^T \mathbf{R}(w)^T\left(\mathbf{u}_i(w) - \mathbf{d}(w)\right)} \\ y_i(w) & = & \dfrac{\mathbf{e}_2^T \mathbf{A}\mathbf{R}(w)^T\left(\mathbf{u}_i(w) - \mathbf{d}(w)\right)}{\mathbf{e}_3^T \mathbf{R}(w)^T\left(\mathbf{u}_i(w) - \mathbf{d}(w)\right)} \end{array}\right. \tag{23}$$

Taking into account the parametrization of $\mathbf{R}(w)$ in (15), one has for $x_i(w)$

$$x_i(w) = \frac{\mathbf{e}_1^T \mathbf{A}\mathbf{\Lambda}(\boldsymbol{\phi}(w))^T\left(\mathbf{u}_i(w) - \mathbf{d}(w)\right)}{\mathbf{e}_3^T \mathbf{\Lambda}(\boldsymbol{\phi}(w))^T\left(\mathbf{u}_i(w) - \mathbf{d}(w)\right)}. \tag{24}$$

As we can see, the denominator in (15) does not affect the image projections, which means that we derive an optimization problem where the rotation is equivalently parameterized by a simple quadratic expression, namely $\mathbf{\Lambda}(\boldsymbol{\phi}(w))$ who depends quadratically on the parameter $\boldsymbol{\phi}(w)$ and, according to (17)–(20), depends quadratically on the parameter $\mathbf{M}$. Let us introduce the polynomials

$$\begin{array}{rcl} a_{i,j}(w) & = & \mathbf{e}_j^T \mathbf{A}\mathbf{\Lambda}(\boldsymbol{\phi}(w))^T\left(\mathbf{u}_i(w) - \mathbf{d}(w)\right), \\ & & i = 1,\ldots,n, \; j = 1,2,3. \end{array} \tag{25}$$

It follows that

$$x_i(w) = \frac{a_{i,1}(w)}{a_{i,3}(w)}, \quad y_i(w) = \frac{a_{i,2}(w)}{a_{i,3}(w)}. \tag{26}$$

Hence, the constraint (22) can be rewritten as

$$\left.\begin{array}{rcl} a_{i,1}(w) - x_{min} a_{i,3}(w) & > & 0 \\ -a_{i,1}(w) + x_{max} a_{i,3}(w) & > & 0 \\ a_{i,2}(w) - y_{min} a_{i,3}(w) & > & 0 \\ -a_{i,2}(w) + y_{max} a_{i,3}(w) & > & 0 \\ a_{i,3}(w) & > & 0 \end{array}\right\} \begin{array}{l} \forall w \in [0,1] \\ \forall i = 1,\ldots,n \end{array} \tag{27}$$

Let us observe that the inequality $a_{i,3}(w) > 0$ has been included in order to ensure that the object remains in front of the camera for the whole trajectory. This allows us to get rid of the denominator and obtain only the polynomial inequalities in (27). Each of these inequalities represents one inequality constraint $g_k(w, \mathbf{M}, \mathbf{N}) > 0$ in (21).

Workspace constraints. Let $\mathcal{F}$ be the current camera frame with pose $\{\mathbf{O}, \mathbf{c}\} \in SE(3)$ with respect to the absolute frame. Due to obstacles present in the scene, the robot cannot reach any position of the three-dimensional space. This means that the camera center $\mathbf{c}$ can assume values in a subset of $\mathbb{R}^3$ only. In our path-planning, the camera center is represented by the translation vector $\mathbf{d}(w)$ as pose with respect to the initial frame $\mathcal{F}^\circ$. Therefore, workspace constraints correspond to constraints on this vector. Now, depending on the information available for the robot control problem we are considering, two kinds of workspace constraints can be taken into account.

The first is the absolute workspace constraint. If the cad model of the object is available, one can calculate the translation of the camera pose $\{\mathbf{R}, \mathbf{t}\}$ which is represented by the vector $\mathbf{d}$ according to Section III-A. This means that one can constraint the optimization problem (21) so that the camera center of the generated trajectories belongs to an absolute set referred to either the initial or the desired camera frame. Since the desired camera frame usually represents a reference location for the robot, we consider for instance the case of this absolute set referred to $\mathcal{F}^*$. Let

$$\mathcal{P}(\mathcal{F}, \mathcal{F}^*) = \{\bar{\mathbf{O}}, \bar{\mathbf{c}}\} \tag{28}$$

be the pose of $\mathcal{F}$ with respect to $\mathcal{F}^*$. The absolute set can be expressed as

$$\mathcal{C} = \left\{\bar{\mathbf{c}} \in \mathbb{R}^3 : \; c_j(\bar{\mathbf{c}}) > 0, \; j = 1,\ldots,n_c\right\} \tag{29}$$

where $c_j : \mathbb{R}^3 \rightarrow \mathbb{R}$ are polynomials. Then, one can constraint the trajectories in (21) so that the camera center belongs to $\mathcal{C}$ by defining n_c functions $g_k(w, \mathbf{M}, \mathbf{N})$ as

$$c_j(\mathbf{R}^T(\mathbf{d}(w) - \mathbf{d})), \; j = 1,\ldots,n_c \tag{30}$$

for some indexes k.

The second is the scaled workspace constraint. If the cad model of the object is not available, one can only calculate the scaled translation $\mathbf{t}_{norm}$ of the camera pose $\{\mathbf{R}, \mathbf{t}\}$ still represented by the vector $\mathbf{d}$ in Section III-A. This means that one can only impose scaled constraints for the camera center $\mathbf{o}$ in the optimization problem (21). In particular, suppose that we want to impose constraints referred to $\mathcal{F}^*$. Then, these constraints have the form (30) but in this case they define a set different from (29) that is given by

$$\mathcal{C}_{norm} = \left\{\bar{\mathbf{c}} \in \mathbb{R}^3 : \; c_j(\|\mathbf{t}\|\bar{\mathbf{c}}) > 0, \; j = 1,\ldots,n_c\right\}. \tag{31}$$

As we can see, this set *depends* on the initial camera pose through the term $\|\mathbf{t}\|$. Although less general than the absolute workspace constraints, the scaled workspace constraints can however model typical situations in which the robot works.

B. Costs

One can consider several cost functions in (21) depending on the specific problem. Some are as follows.

Spanned image area. Here we consider the problem of minimizing the area spanned by the image trajectories. This can be done defining the cost function $h(\mathbf{M}, \mathbf{N}) = \Delta$ where

$$\Delta = (\sigma_{x,max} - \sigma_{x,min})(\sigma_{y,max} - \sigma_{y,min}) \tag{32}$$

where $\sigma_{x,min}, \sigma_{x,max}, \sigma_{y,min}, \sigma_{y,max}$ denote the coordinates of the boundary box of the image trajectory according to

$$\begin{cases} \sigma_{x,min} = \min\{x_i(w),\ w \in [0,1],\ i = 1,\dots,n\}, \\ \sigma_{x,max} = \max\{x_i(w),\ w \in [0,1],\ i = 1,\dots,n\}, \\ \sigma_{y,min} = \min\{y_i(w),\ w \in [0,1],\ i = 1,\dots,n\}, \\ \sigma_{y,max} = \max\{y_i(w),\ w \in [0,1],\ i = 1,\dots,n\}. \end{cases} \tag{33}$$

In order to compute these coordinates, let us consider first $\sigma_{x,min}$. Since $x_i(w)$ in (26) is a rational function whose denominator is positive for all $w \in [0,1]$, one can compute $\sigma_{x,min}$ by evaluating each $x_i(w)$ at the extremes point of the trajectory and at the points in $[0,1]$ where its derivative vanishes:

$$\begin{cases} \sigma_{x,min} &=& \min\{x_{i,min},\ i = 1,\dots,n\} \\ x_{i,min} &=& \min\{\frac{a_{i,1}(0)}{a_{i,3}(0)}, \frac{a_{i,1}(1)}{a_{i,3}(1)}, \bar{x}_{i,min}\} \\ \bar{x}_{i,min} &=& \min\{\frac{a_{i,1}(w)}{a_{i,3}(w)},\ w:\ \frac{\partial a_{i,1}(w)}{\partial w} a_{i,3}(w) \\ & & -a_{i,1}(w)\frac{\partial a_{i,1}(w)}{\partial w} = 0\} \end{cases} \tag{34}$$

Therefore, the computation of $\sigma_{x,min}$ requires just the computation of the roots of a one-variable polynomial. The other coordinates $\sigma_{x,max}, \sigma_{y,min}, \sigma_{y,max}$ are analogously calculated.

Trajectory length. Another useful cost is the length of the camera trajectory in the three-dimensional space, which can be imposed as

$$h(\mathbf{M},\mathbf{N}) = \int_0^1 \left\| \frac{\partial \mathbf{d}(w)}{\partial w} \right\| dw \tag{35}$$

The integral can be computed through finite difference approximations in order to speed up the calculation.

V. Simulations in ideal and real conditions

In this section we present some examples of the proposed approach through simulations in ideal conditions and simulations in real conditions, that is in the presence of image noise, uncertainties on the intrinsic parameters and uncertainties on the extrinsic parameters.

Once the problem (21) has been solved, the planned image trajectories corresponding to the optimal values of $\mathbf{M}$ and $\mathbf{N}$ are computed, and then these image trajectories are tracked by using an IBVS controller similar to that proposed in [11]. We suppose that the screen size is 800×600 pixels and that the intrinsic parameters are $\mathbf{A} = [400, 0, 400; 0, 400, 300; 0, 0, 1]$. The real conditions are characterized as follows.

- (RC1) Image noise: each image projection is randomly shifted, with uniform distribution, in a square with side equal to 1 pixel centered on the point itself.
- (RC2–RC3) Calibration errors: the matrices $\mathbf{J}$ (robot Jacobian, necessary to implement the IBVS controller) and $\mathbf{A}$ are supposed coarsely estimated by the estimates

$$\hat{\mathbf{J}} = \begin{bmatrix} \mathbf{R}_E & [\mathbf{t}_E]_\times \mathbf{R}_E \\ \mathbf{0}_{3\times 3} & \mathbf{R}_E \end{bmatrix} \mathbf{J},\ \hat{\mathbf{A}} = \begin{bmatrix} 385 & 0 & 408 \\ 0 & 402 & 294 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\mathbf{t}_E \in \mathbb{R}^3$ and $\mathbf{R}_E \in SO(3)$ accounts for errors in the coordinates transformation and are selected as

$$\mathbf{t}_E = [2, -0.5, 0]^T \text{ cm},\ \mathbf{R}_E = e^{\left[\pi[0.01,0,0.02]^T\right]_\times}.$$

- (RC4) Non-perfect IBVS control: the point depths in the current camera frame is supposed not exactly known. In particular, we select to use the point depths in the virtual camera frame $\mathcal{F}_d(w(t))$ in order to lighten the on-line computational burden.

The problem (21) is solved for simple polynomials $\boldsymbol{\phi}(w)$ and $\mathbf{d}(w)$ in (17) of degree 2. Once the image trajectories are computed, the IBVS control law is applied by selecting w equal to $w = 1 - e^{-t/\tau}$ where $\tau = 10$ s.

Consider the situation depicted in Figure 1a where a set of twelve balls represents the object observed by the camera in the initial and desired frames $\mathcal{F}^\circ$ and $\mathcal{F}^*$. A horizontal plane with equation $y = 15$ cm is inserted under the object to limit the robot workspace. Figure 1b shows the centers of the balls in both initial and desired camera views. It is supposed that no cad model of the object is available.

The problem consists of steering the camera from $\mathcal{F}^\circ$ to $\mathcal{F}^*$ minimizing the spanned image area. Figures 1c–d show the results obtained without considering the workspace constraint imposed by the presence of the plane. As we can see, the trajectory of the camera goes under the plane. This kind of problems can easily happen in visual servoing if the workspace constraints are not taken into account.

In order to take into account this constraint, we can proceed as described in Section IV. In particular, if a cad model is available we can consider the presence of the plane by simply defining an absolute workspace constraint as $\mathcal{C} = \left\{\bar{\mathbf{c}} \in \mathbb{R}^3 :\ \bar{c}_2 + 15 \geq 0\right\}$. However, we suppose that a cad model of the object is not available. This means that we can describe the distance of the plane from the desired camera frame only up to a scale factor. Therefore, the only constraint we can impose is that the camera does not go below the horizontal plane passing through the initial and desired camera frames. This is achieved with the scaled workspace constraint $\mathcal{C} = \left\{\bar{\mathbf{c}} \in \mathbb{R}^3 :\ \bar{c}_2 \geq 0\right\}$. Figures 1e–f show the results obtained in ideal conditions, whereas Figures 1g–j show the results in the real conditions RC1–RC4. As we can see, the trajectory followed by the camera is almost horizontal in spite of the uncertainties, and clearly avoids the problem introduced by the workspace limitations.

Let us observe that, in the presence of image noise and calibration errors, the real image trajectories will differ from the planned ones, especially if a cad model of the object is not available. In order to guarantee that the constraints are however satisfied, one can consider more conservative constraints in the planning phase in order to facilitate their fulfillment by the real camera trajectory. Also, one can add to the IBVS control law potential field techniques similar to that used in [2] (a potential field is used on-line for the visibility constraint) and [11] (potential fields are used off-line to plan the camera trajectory). In our case, we can introduce an *on-line potential field for each constraint.*

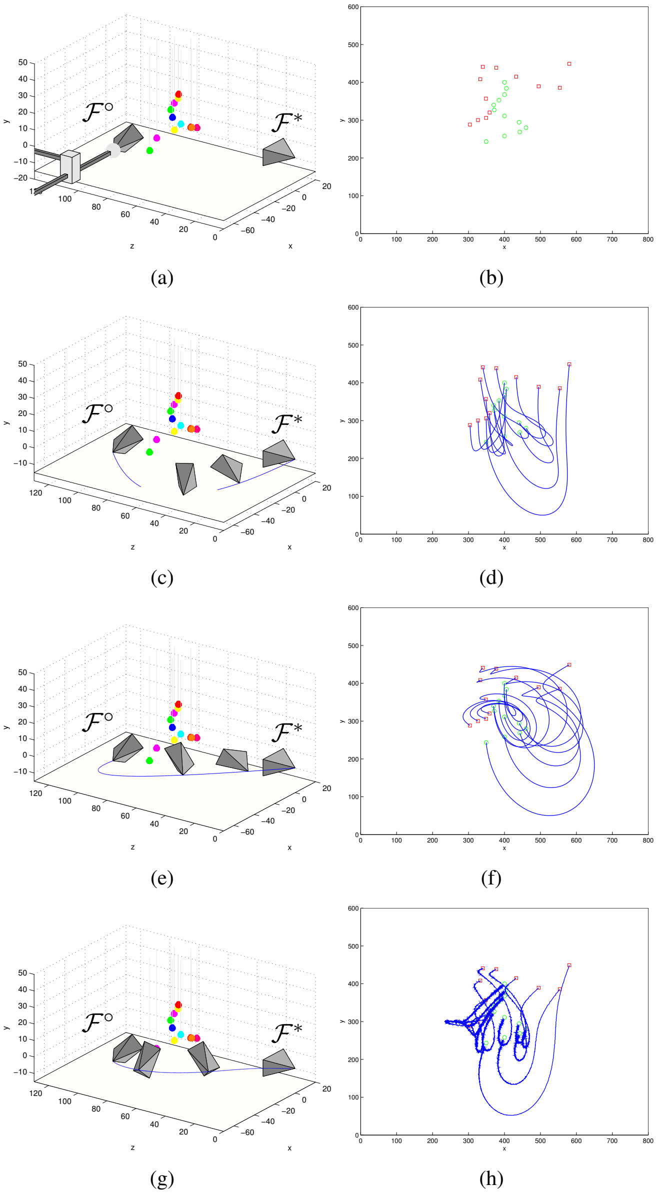

Fig. 1. Example. (a-b) Initial configuration. (c-d) Results in ideal conditions: the camera goes under the plane. (e-f) Results in ideal conditions taking into account the workspace constraint. (g-h) Results in real conditions.

VI. Conclusion

This paper has proposed a new path-planning scheme for constrained and optimal visual servoing. In particular, all the trajectories connecting the initial to the desired location are parameterized in the six-dimensional rigid motion space. This is achieved by introducing a new robust object reconstruction which allows one to obtain image trajectories satisfying the boundary conditions even in the presence of calibration errors and image noise. In order to obtain functions that can be efficiently handled in optimization tools, the rotation path is parameterized through a particular extension of the Euler parameters which allows one to obtain an equivalent expression of the rotation as a quadratic function of unconstrained variables. Polynomials of arbitrary degree are then used to complete the parametrization and formulate a general optimization where a number of constraints such as visibility and workspace constraints, and a number of costs such as spanned image area and trajectory length, can be considered. Thanks to the introduced trajectory parametrization, establishing the fulfillment of the constraints in the optimization procedure reduces to the simple calculation of the roots of a one-variable polynomial, hence largely simplified with respect to standard parameterizations which involve transcendental functions. Once the image trajectories have been computed, the camera is steered to the desired location by using standard IBVS controllers.

References

[1] G. Chesi and A. Vicino. Visual servoing for large camera displacements. *IEEE Trans. on Robotics*, 20(4):724–735, 2004.

[2] P.I. Corke and S.A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Trans. on Robotics and Automation*, 17(4):507–515, 2001.

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

[4] O. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images*. MIT Press, Cambridge (Mass.), 2001.

[5] K. Hashimoto. A review on vision-based control of robot manipulators. *Advanced Robotics*, 17(10):969–991, 2003.

[6] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura. Manipulator control with image-based visual servo. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2267–2272, 1991.

[7] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, 1996.

[8] E. Malis. Visual servoing invariant to changes in camera-intrinsic parameters. *IEEE Trans. on Robotics and Automation*, 20(1):72–81, 2004.

[9] E. Malis and F. Chaumette. 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *Int. Journal of Computer Vision*, 37(1):79–97, 2000.

[10] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 D visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, 1999.

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

[12] Y. Mezouar and F. Chaumette. Optimal camera trajectory with image-based control. *Int. Journal of Robotics Research*, 22(10–11):781–803, 2003.

[13] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, 1994.

[14] P. Oh and P. Allen. Visual servoing by partitioning degrees-of-freedom. *IEEE Trans. on Robotics and Automation*, 17(1):1–17, 2001.

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

[16] A. C. Sanderson, L. E. Weiss, and C. P. Neuman. Dynamic sensor-based control of robots with visual feedaback. *IEEE Trans. on Robotics and Automation*, RA-3:404–417, 1987.

[17] O. Tahri and F. Chaumette. Image moments: generic descriptors for decoupled image-based visual servo. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1185–1190, New Orleans, Louisiana, 2004.

[18] C.J. Taylor and J.P. Ostrowski. Robust vision-based pose control. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2734–2740, San Francisco, California, 2000.

[19] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice. Position based visual servoing: keeping the object in the field of vision. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1624–1629, Washington, D.C., 2002.

[20] H. Zhang and J.P. Ostrowski. Visual motion planning for mobile robots. *IEEE Trans. on Robotics and Automation*, 18(2):199–208, 2002.