Robot Positioning via Image Based Visual Servoing

Duccio Fioravanti[1] , Benedetto Allotta[2] , Rindi Andrea[3]
Department of Energetics, University of Florence, Italy
E-mail: [1]*fioravanti@mapp1.de.unifi.it,* [2]*benedetto.allotta@unifi.it,* [3]*andrea.rindi@unifi.it*

Keywords: Image Based Control, Path Planning, Homography.

SUMMARY. Visual Servoing has become a popular paradigm for the control of complex robotic systems: this sensor based approach exploits the image informations provided by one ore more cameras in a feedback control loop to drive the system to the desired configuration. Here authors will refers to a monocular system where the camera is mounted on the end effector af a 6-DOF manipulator. Among different Visual Servoing approaches Image Based Visual Servoing (IBVS) has been the most investigated in the literature because of its nice properties of robustness with respect to both robot modeling and camera calibration errors: in IBVS the control loop is in fact directly closed in the image; moreover IBVS doesn't require the knowledge of the target/scene model (model-free approach). Despite its advantages IBVS may be affected by singularity and local minima problems of the control law: these drawbacks arise especially when the initial and the goal camera images respectively corresponding to the actual and desired system configurations are very different (i.e for large system displacements). To overcome these problems an image path planning can be exploited to ensure system convergence. In this paper author presents an off-line image path planning that can be used to execute system positioning task also in presence of large camera displacements: planning trajectories has been developed such as to make the robot end effector move on a 3D helix, connecting the initial and the desired arm configuration, by generating feasible robot twist-screws and keeping the target in the image field of view. During control execution also 3D target informations are retrieved through an adaptive estimation law. Experimental results with a real robot show the feasibility of the proposed approach.

1 INTRODUCTION

Nowadays the choice of vision sensors to control in real-time complex robotic systems has become a common use, thanks to the increasing performances of both modern computers and hardware components. Various techniques have been proposed to control the behavior of a dynamic system exploiting the visual information provided by one ore more cameras inside a feedback control-loop: these approaches are known in literature as Visual Servoing [Hutchinson et al., 1996].
Using Visual Servoing a robot can be positioned with respect to a target placed in its workspace, by minimizing the differences between the actual target view (correspondent to an actual robot configuration) and the desired camera view (correspondent to the desired robot configuration). Visually controlled systems may differ in the control architecture (direct and indirect visual servoing), in the number of camera used (mono, stereo or multi-camera configurations) and in their placing in the system (eye-in-hand or eye-to-hand systems).
Concerning the features of the control scheme, Visual Servoing approaches can be subdivided into three different groups, depending on the definition of the error function adopted to regulate the system: Position Based Visual Servoing (PBVS), Image based Visual Servoing (IBVS) and Hybrid Visual Servoing.
In Position Based Visual Servoing the error is computed by estimating from images some 3D features of interest (like the desired camera rotation and translation etc...): this approach allows the user

to fully define the robot trajectory in 3D space but has proved to be very sensitive with respect to both camera and robot calibration errors and usually requires the knowledge of the $3D$ target model [Hutchinson et al., 1996].
In Image based Visual Servoing, on the other hand, the feedback control loop is directly closed in the image since the system error is defined by primitives directly extracted from images (such as 2D points, lines, or more complex shapes [Espiau et al., 1992]).
Image based approach in general doesn't require a priori knowledge of the target (model-free technique) and is known to be really robust with respect to both camera and robot calibration errors [Espiau, 1993]. On the other hand only local stability of the controlled system has been proved: that is why, during task executions, singularities and local minima of the control law may occur especially when the initial and the desired view of the target are noticeably different (large camera displacements) [Chaumette, 1998]. To overcome singularity and local minima problems a convenient path planning in the image can be exploited.
The hybrid visual servoing scheme uses an error defined on both image and Cartesian primitives: this method ensures better analytical stability properties (sufficient conditions for global stability has been proved) [Malis et al., 1999]; however also in this approaches 3D robot trajectory is not predictable and the target may go out of the camera field of view, causing the loss of the visual informations (the visual features) and thus the failure of the task.
In this paper authors report experimental results regarding the execution of positioning tasks with respect to a fixed planar target using a 6 DOF manipulator with a single camera mounted on its end-effector (eye-in-hand system). Positioning task are fulfilled using IBVS, by the generating of an opportune image path-planning so that the robot end-effector is driven to the goal pose following an helical trajectory in 3-D space. The image generated paths give rise to feasible camera velocity-screws compatible with the robot actuators and maintain the target in the camera field of view. The paper is structured as follows: Section (2) describes the camera model used and the projective geometry relations between two views used in the work; Section (3) addresses the control problem and the image planning generation while Section (4) shows experimental results; finally conclusion and future work are reported in Section (5).

2 MODELLING

First to address the control problem, it is convenient to define the geometric projection model of the camera sensor (mapping the Euclidean world into a digital image); by this way, the Projective Geometry theory can be exploited to describe the relation between two different images (i.e. the actual and the desired views) of a 3D target. The results obtained in this section will be useful in the following of the paper.

2.1 Camera Model

As shown in Figure (1 a)), the camera can be properly modeled by a frame attached to the sensor itself defined as *camera frame* (having its z axis coincident with the focal axis and origin in the projection center) and by an upper triangular 3×3 matrix $\boldsymbol{K}$ embedding the projection model (known in literature as intrinsic camera parameters matrix)[Hartley and Zisserman, 2003]. Using the *full projective camera* model we have:

$$\boldsymbol{K} = \begin{bmatrix} \alpha_x & \sigma & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where α_x, α_y are the *focal distance* f expressed in pixels respectively along the pixel base and height, σ is the *skew factor* and x_0,y_0 are the *principal point* pixels coordinates. In this work $\boldsymbol{K}$ is supposed to be known and assumed to be constant (no-zoom). Let be $\boldsymbol{P_c} = [X\,Y\,Z]^T$ the $3D$

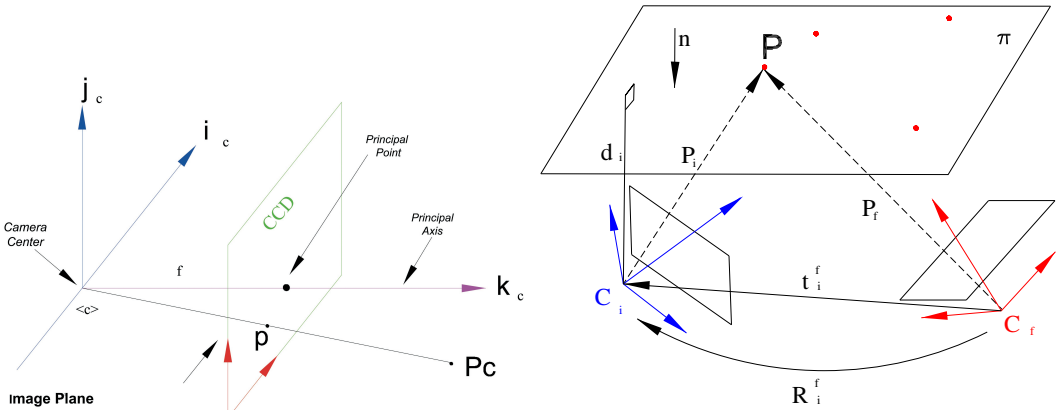


Figure 1: a) Finite projective CCD camera model; b) Initial view I_i and final view I_f of a Planar target.

coordinates of a generic target point with respect to the camera frame; we define the *normalized image point* (in homogeneous coordinates) as:

$$\tilde{\boldsymbol{m}} = \left[\begin{array}{cc} \boldsymbol{m}^T & 1 \end{array}\right]^T = \left[\begin{array}{ccc} x & y & 1 \end{array}\right]^T = \frac{1}{Z}\boldsymbol{P_c}\,. \tag{2}$$

From (2) we can define the pixel coordinates of the same point by the following relation:

$$\tilde{\boldsymbol{p}} = \left[\begin{array}{ccc} u & v & 1 \end{array}\right]^T = \boldsymbol{K}\tilde{\boldsymbol{m}}\ . \tag{3}$$

Since $\boldsymbol{K}$ is known, it is straightforward to find the normalized coordinates $\boldsymbol{m}$ from the known pixel coordinates $\tilde{\boldsymbol{p}}$ by inverting (3) for each image target point of interest.

2.2 Two Views Projective Geometry

Consider an initial and a final view I_i and I_f of a planar target corresponding respectively to the initial camera frame C_i and to the final camera frame C_f, as shown in Figure (1 b)); let be $\boldsymbol{P}$ the generic target point belonging to the target plane π, $\boldsymbol{R}_{fi}$ and $\boldsymbol{t}_{fi}$ the orthogonal rotation matrix and the translation vector between C_f and C_i. We have:

$$\boldsymbol{P}_f = \boldsymbol{R}_{fi}\boldsymbol{P}_i + \boldsymbol{t}_{fi} \quad \Rightarrow \quad \frac{Z_f}{Z_i}\boldsymbol{m}_f = \boldsymbol{R}_{fi}\boldsymbol{m}_i + \boldsymbol{t}_{fi}\frac{1}{Z_i} \quad , \tag{4}$$

where the right equation has obtained by using eq.(2). Since the point $\boldsymbol{P} \in$ to the plane π, the following relation can be written:

$$\boldsymbol{n}_i^T\boldsymbol{P}_i = d_i \Rightarrow 1/Z_i = (\boldsymbol{n}_i^T\boldsymbol{m}_i)/d_i \quad . \tag{5}$$

Injecting eq.(5) in eq.(4) we obtain:

$$\frac{Z_f}{Z_i}\boldsymbol{m}_f = \boldsymbol{R}_{fi}\boldsymbol{m}_i + \boldsymbol{t}_{fi}\frac{\boldsymbol{n}_i^T\boldsymbol{m}_i}{d_i} \quad \Rightarrow \quad \frac{Z_f}{Z_i}\boldsymbol{m}_f = \left(\boldsymbol{R}_{fi} + \boldsymbol{t}_{fi}\frac{\boldsymbol{n}_i^T}{d_i}\right)\boldsymbol{m}_i = \boldsymbol{H}_{fi}\boldsymbol{m}_i \quad , \tag{6}$$

The matrix $\boldsymbol{H}_{fi} \in 3 \times 3$ is known as Euclidean Homography: it defines the mapping between two images of the same planar target with image point expressed in normalized coordinates. $\boldsymbol{H}_{fi}$ can also be written as:

$$\boldsymbol{H}_{fi} = \quad \left(\boldsymbol{R}_{fi} + \boldsymbol{ts}_{fi}\boldsymbol{n}_i^T\right) \quad = \quad {\boldsymbol{R}_{if}}^T(I - \boldsymbol{ts}_{if}{\boldsymbol{n}_i}^T) \quad , \tag{7}$$

where $\boldsymbol{ts}_{if} = (\boldsymbol{t}_{if}/d_i)$ is the scaled translation between C_i and C_f . Notice that in the right member of eq.(7) all terms are expressed with respect to C_i. Eq.(6), transformed in pixel coordinates, becomes:

$$\tilde{\boldsymbol{p}}_f \propto \boldsymbol{G}_{fi}\tilde{\boldsymbol{p}}_i \quad ; \tag{8}$$

$\boldsymbol{G}_{fi}$ represents a linear mapping between the I_i and I_f. Since we assume K constant we have :

$$\boldsymbol{G}_{fi} = \boldsymbol{K}\boldsymbol{H}_{fi}\boldsymbol{K}^{-1} \quad . \tag{9}$$

The matrix $\boldsymbol{G}_{fi}$ is the homography between two views of the same planar target in pixel coordinates; notice that if C_i and π are fixed, both $\boldsymbol{G}_{fi}$ and $\boldsymbol{H}_{fi}$ have only 6 DOF.
Suppose the unknown fixed target to be identified in each image by four well detectable points assumed to be coplanar but not collinear, represented respectively in I_i and I_f by $\widetilde{\boldsymbol{p}}_j^i = [u_j^i v_j^i 1]^T$ and $\widetilde{\boldsymbol{p}}_j^f = [u_j^f v_j^f 1]^T$, with $j = 1...4$. Exploiting the knowledge of $\widetilde{\boldsymbol{p}}_j^i$, $\widetilde{\boldsymbol{p}}_j^f$ and of their correspondences between I_i and I_f, we apply the following procedure:

1. Compute $\boldsymbol{G}_{fi}$ (there are several methods, see [Hartley and Zisserman, 2003]).
2. Retrieve $\boldsymbol{H}_{fi}$ up to a scalar factor, by inversion of eq.(8), since we assume to know K.
3. Compute $\boldsymbol{H}_{fi}$ (using the algorithm expressed in [Ma et al., 2003]) and decompose this homography in its rotational ad translational components, obtaining four different sets of solutions for the variables $\boldsymbol{R}_{if}$, $\boldsymbol{ts}_{if}$ and $\boldsymbol{n}_i$.
4. Find the *real* set of solutions from the four given candidates (for this we need an auxiliary view defined by I_a, that differs from the other two at least for a translation).

From the images I_i, I_f,I_a and $\boldsymbol{K}$, we are finally able to determine the rotation and the translation up to the scalar factor d_i between the camera frames C_f and C_i, obtaining a scaled Euclidean reconstruction of the unknown planar target up to the scalar d_i factor.

3 CONTROL

To build up and appropriate control law for the system, let us first define the relation between the velocity-screw applied to the camera frame (through robot actuators) and the feature-points velocities in the image. Consider for a generic feature point the normalized coordinates $\boldsymbol{m}$ obtained from $\tilde{\boldsymbol{p}}$ by inversion of eq.(3); let be $\boldsymbol{m}_i$ the non-homogeneous normalized coordinates (2) of the generic feature point :

$$\boldsymbol{m}_i = \begin{bmatrix} X_i/Z_i \\ Y_i/Z_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{10}$$

Let be $\boldsymbol{w}$ the Camera Twist-Screw:

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = [v_x\, v_y\, v_z\, \omega_x\, \omega_y\, \omega_z]^T \quad , \tag{11}$$

where $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are linear and angular camera velocity expressed in the its frame C. The velocity of a $3D$ target point $\boldsymbol{P}_c$ with respect to C is linked to $\boldsymbol{w}$ by:

$$\dot{\boldsymbol{P}}_c = -\boldsymbol{v} - \boldsymbol{\omega} \wedge \boldsymbol{P}_c \qquad . \tag{12}$$

Making the time derivate of (10) and using (12), we obtain the image jacobian for an normalized image point $\tilde{\boldsymbol{m}}_i$:

$$\dot{\boldsymbol{m}}_i = \begin{bmatrix} -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i y_i & -\left(1+x_i^2\right) & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & \left(1+y_i^2\right) & -x_i y_i & -x_i \end{bmatrix} \boldsymbol{w} = J_i \boldsymbol{w}, \tag{13}$$

with $\boldsymbol{m}_i = [x_i\, y_i]^T$ (in non-homogenous coordinates). J_i is called *Interaction Matrix* or Image Jacobian of the generic feature point $\boldsymbol{P}_i = [X_i\, Y_i\, Z_i]^T$. The Interaction Matrix represents a local linear mapping between the feature point velocity in the image and the camera twist-screw: notice that J_i depends on the point image coordinates x_i,y_i and on the $3D$ point depth Z_i with respect to C (variable with camera pose). The unknown depth Z_i are on-line estimated in the control phase by the adaptive law reportrd in [Conticelli and Allotta, 2001].

3.1 System Dynamic

Using in general n points to define the target, we can build the system state vector:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{m}_1^T & \boldsymbol{m}_2^T & \dots & \boldsymbol{m}_n^T \end{bmatrix}^T \quad , \tag{14}$$

from (13), stacking the interaction matrix for all the image points, the global image jacobian $J \in \boldsymbol{R}^{2n\times 6}$ results:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1^T & \boldsymbol{J}_2^T & \dots & \boldsymbol{J}_n^T \end{bmatrix}^T \quad . \tag{15}$$

From (14) and from (15), we obtain the dynamics of the system to be controlled:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{x}, \boldsymbol{Z_x})\boldsymbol{w}. \tag{16}$$

3.2 Control law

Let us define the image error (system error) as the difference between the reference and the actual normalized image points:

$$\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x} \quad . \tag{17}$$

By inspection of (16), we can use a control law based on the image error $\boldsymbol{e}$ and on the pseudoinverse of the estimate jacobian $\hat{J}^{\dagger}$:

$$\boldsymbol{w} = \hat{\boldsymbol{J}}^{\dagger}(k\boldsymbol{e} + \dot{\boldsymbol{x}}_d) \quad , \tag{18}$$

where $\hat{\boldsymbol{J}}^{\dagger} = (\hat{\boldsymbol{J}}^T \hat{\boldsymbol{J}})^{-1} \hat{\boldsymbol{J}}^T$ and k is a positive gain. Expression (18) can be used only when $\hat{J}$ has full rank . Notice that the vector $\boldsymbol{x}_d$ represents the reference feature set: if the IBVS control doesn't provide any image planning it corresponds with the desired view features set $\Rightarrow \boldsymbol{x}_{(d)} = \boldsymbol{x}_{(f)} = cost$; on the other hand when path planning is done $\boldsymbol{x}_{(d)} \neq cost$ corresponds with the planned image trajectories ranging from $\boldsymbol{x}_i$ to $\boldsymbol{x}_f$.

3.3 System Stability

To prove the stability of the system we use the Lyapunov candidate:

$$V = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} \quad . \tag{19}$$

The time derivate of (19), by using (18) and (16) results:

$$\begin{aligned} \dot{V} &= \boldsymbol{e}^T\dot{\boldsymbol{e}} = \boldsymbol{e}^T(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) = \boldsymbol{e}^T(\dot{\boldsymbol{x}}_d - \boldsymbol{J}\boldsymbol{w}) \ , \\ \dot{V} &= \boldsymbol{e}^T(\dot{\boldsymbol{x}}_d - \boldsymbol{J}\hat{\boldsymbol{J}}^\dagger(k\boldsymbol{e} + \dot{\boldsymbol{x}}_d)) \ , \\ \dot{V} &= -k\boldsymbol{e}^T J\hat{\boldsymbol{J}}^\dagger\boldsymbol{e} + \boldsymbol{e}^T(\boldsymbol{I} - \boldsymbol{J}\hat{\boldsymbol{J}}^\dagger)\dot{\boldsymbol{x}}_d \quad . \end{aligned} \tag{20}$$

If a good estimate of $\boldsymbol{J}$ is computed, $(\boldsymbol{e}^T\boldsymbol{J}\hat{\boldsymbol{J}}^\dagger\boldsymbol{e})$ in the first term is semidefinite positive while the matrix $(I - J\hat{J}^\dagger)$ of the second term is a projector in the null space of $\boldsymbol{J}^T$: thus, if $\boldsymbol{e}$ or $\dot{\boldsymbol{x}}_d \ \notin N(\boldsymbol{J}^T)$ the second term vanishes, we have $\dot{V} \leq 0$ and system local stability is proved. However, especially when I_i and I_f are very different (large camera displacements), system convergence is not ensured (see [Hutchinson et al., 1996]):

1. *Singularities* could occur when rank J is not full ($\mathrm{rank}(\boldsymbol{J}) < 6$)
2. *Local Minima* may arise when the image error $\boldsymbol{e} = (\boldsymbol{x}_d - \boldsymbol{x})$ is not zero and belongs to $\mathcal{N}(\boldsymbol{J}^T)$ (null space of $\boldsymbol{J}^T$), even if $\boldsymbol{J}$ has full rank ($\mathrm{rank}(\boldsymbol{J}) = 6$). Local minima occur when image reference trajectory of feature point tends to unrealizable images.

In first case control becomes unstable, in the second system converges to configurations different from the goal. To overcome these drawbacks is essential to keep $\boldsymbol{e}$ as small as possible by performing a path planning on the reference image features $\boldsymbol{x}_d$ taking also into account the rigid displacement of the camera in $3D$ Euclidean space.

3.4 The Planning Method

To perform image path planning, Euclidean Homography Structure $\boldsymbol{H}$ can be exploited: since between I_i and I_f we have $\tilde{\boldsymbol{m}}_f \propto H_{fi}\tilde{\boldsymbol{m}}_{\boldsymbol{i}}$ (see (6)), planned image trajectories $\boldsymbol{x}_d$ can be generated defining a *Time-varying Reference Homography* $\boldsymbol{H}_d(s)$ such that:

$$\tilde{\boldsymbol{m}}_d(s) \propto \boldsymbol{H}_d(s)\tilde{\boldsymbol{m}}_{\boldsymbol{i}} \Rightarrow \boldsymbol{x}_d(s), \tag{21}$$

where $s = s(t)$ is a suitable time law. Camera-target twist-screw can thus be generated smoothly varying $\boldsymbol{H}_d(s)$ between the actual and the desired views from $\boldsymbol{I}$ to $\boldsymbol{H}_{fi}$. Let summarize the key points our approach:

1. Define a planning Homography $\boldsymbol{H}_d(s)$ from the recovered camera kinematic parameters $\boldsymbol{R}_{if}$ $\boldsymbol{ts}_{if}$ and $\boldsymbol{n}_i$ (see section (2.2)), where $s(t)$ is an opportune time law.
2. Generate the planned image trajectories $\boldsymbol{x}_d$ for the 4 feature points by exploiting $\boldsymbol{H}_d(s)$ to compute image error $\boldsymbol{e}$.
3. Regulate the system error $\boldsymbol{e} = (\boldsymbol{x}_d - \boldsymbol{x})$ between actual and planned points coordinates.
4. Perform an adaptive estimation of actual feature point depths $Z_i \quad i = 1...4$ to build Interaction Matrix $\boldsymbol{J}$.

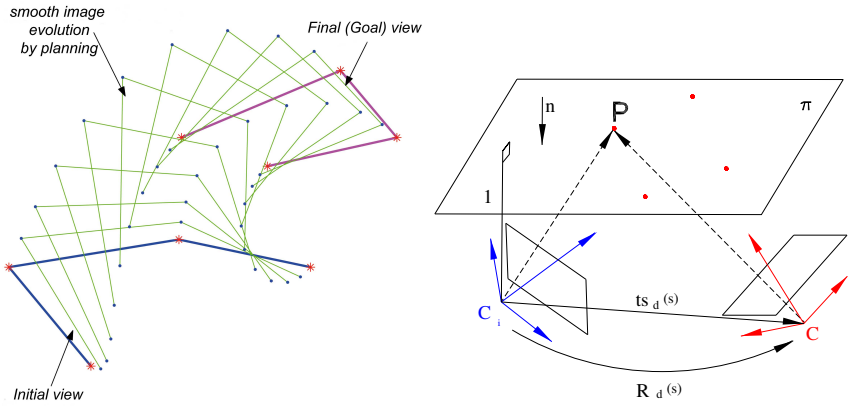


Figure 2: Left: path planning in the image. Right: reference camera rotation and translation used to generate $\boldsymbol{H}_d(s)$.

3.5 Image Path Planning

Recall that we have:

$$\tilde{\boldsymbol{m}}_d(s) \propto \boldsymbol{H}_d(s)\tilde{\boldsymbol{m}}_{\boldsymbol{i}} \quad \Rightarrow \quad \boldsymbol{x}_d(s) = \boldsymbol{f}(\boldsymbol{H}_d(s)). \tag{22}$$

$s = s(t)$, with $(0 \leq s \leq 1)$, is a scalar time law such that $\boldsymbol{H}_d(s)$ verifies the following conditions:

$$\boldsymbol{H}_d(0) = \boldsymbol{I} \quad \boldsymbol{H}_d(1) = \boldsymbol{H}_{fi} \tag{23}$$

To obtain smooth variations of the reference image features we define $s(t)$ as a quintic-polynomial:

$$s(t/t_f) = a(t/t_f)^5 + b(t/t_f)^4 + c(t/t_f)^3 + d(t/t_f)^2 + e(t/t_f) + f \quad . \tag{24}$$

The coefficients in s can be found satisfying the following conditions:

$$s(0) = 0, \qquad \dot{s}(0) = 0, \qquad \ddot{s}(0) = 0, \qquad s(t_f) = 1, \qquad \dot{s}(t_f) = 0, \qquad \ddot{s}(t_f) = 0 \quad . \tag{25}$$

To plan $\boldsymbol{H}_d(s)$ compatible with the rigid camera motion, we exploit Euclidean Homography structure (see 7):

$$\boldsymbol{H}_d(s) = \boldsymbol{R}_d(s)^T(\boldsymbol{I} - \boldsymbol{ts}_d(s)\boldsymbol{n}_i{}^T) \quad ; \tag{26}$$

where to verify equation (23) we have to impose:

$$\boldsymbol{R}_d(0) = I, \quad \boldsymbol{R}_d(1) = \boldsymbol{R}_{if}, \quad \boldsymbol{ts}_d(0) = \boldsymbol{0}, \quad \boldsymbol{ts}_d(1) = \boldsymbol{ts}_{if} \quad . \tag{27}$$

The definition of $\boldsymbol{H}_d(s)$ is now split into two sub-problems: the definition of $R_d(s)$ (reference rotation) and of $\boldsymbol{ts}_d(s)$ (reference scaled translation). The reference image features $\boldsymbol{x}_d$ will be finally computed by (22). Rotation matrix $R_d(s)$ is computed by Rodriguez formula:

$$\boldsymbol{R}_d(s) = \boldsymbol{I} + [\boldsymbol{u}]\sin(\theta_{if}s) + [\boldsymbol{u}]^2(1 - \cos(\theta_{if}s)) \tag{28}$$

where $\boldsymbol{u}$ is the finite rotation axis (unit vector) between C_i and C_f, θ_{if} is the overall rotation angle and $[\,]$ denotes skew operator. $\boldsymbol{u}$ and θ_{if} are computed from the overall rotation $\boldsymbol{R}_{if}$ obtained in

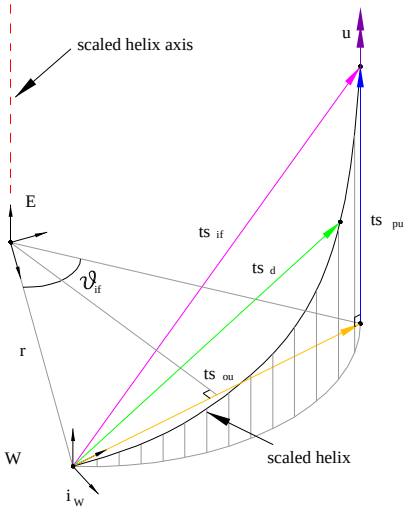


Figure 3: scaled helix defined by $\boldsymbol{R}_d(s)$ and $\boldsymbol{ts}_d(s)$.

Section (2.2) through its axis-angle representation. The reference scaled translation $\boldsymbol{ts}_d(s)$ assigns the $3D$ *camera trajectory shape* followed during task execution: we choose in this work an helical trajectory since it well harmonizes camera rotation and translation. The "scaled" helix has the form:

$$\boldsymbol{ts}_{d(E)} = \begin{cases} x(s) = r\cos(\theta_{if}s)) \\ y(s) = r\sin(\theta_{if}s) \\ z(s) = \frac{p}{2\pi}\theta_{if}s \end{cases} \tag{29}$$

where the above are the coordinates with respect to a canonical frame E (in a scaled space such that $d_i = 1$): E has z axis parallel to the unit vector $\boldsymbol{u}$; the overall helix angle is equal to θ_{if} (Fig.3). p and r are the pitch and the radius of the scaled helix computed by $\boldsymbol{ts}_{if}$ and $\boldsymbol{u}$. Let be in fact:

$$\boldsymbol{ts}_{pu} = (\boldsymbol{ts}_{if} \bullet \boldsymbol{u})\boldsymbol{u}, \qquad \boldsymbol{ts}_{ou} = \boldsymbol{ts}_{if} - \boldsymbol{ts}_{pu} \quad , \tag{30}$$

respectively the parallel and orthogonal components of the overall scaled translation $\boldsymbol{ts}_{if}$ with respect to the unit vector $\boldsymbol{u}$; we have:

$$p = \frac{2\pi}{\theta_{if}}({\boldsymbol{ts}_{if}}^T \bullet \boldsymbol{u}), \qquad r = \frac{|\boldsymbol{ts}_{ou}|}{2\sin(\theta_{if}/2)} \quad . \tag{31}$$

From $\boldsymbol{ts}_{d(E)}$, $\boldsymbol{ts}_d$ is easily computed by a change of coordinates. Even if distance d_i is unknown (see Section 2.2), vector $\boldsymbol{ts}_d$ is completely defined. When $\boldsymbol{ts}_{pu} = \boldsymbol{0}$, $p = 0$ and the helix degenerates into an arc; when $\boldsymbol{ts}_{ou} = \boldsymbol{0}$, $r = 0$ and the helix becomes a line-segment. To suit helical path planning to every possible task, we suggest two "optional" modifications (see Figure 4). In order to avoid that helix convexity turns toward the target plane π , if $(\boldsymbol{i}_W \bullet \boldsymbol{n}_i > 0)$, we plan $\boldsymbol{ts}_d$ onto a new helix symmetric to the old one with respect to the coordinate plane of W with unit normal $\boldsymbol{i}_W$. The second modification allows to plan $\boldsymbol{ts}_d$ onto a new helix characterized by the following parameters: $\boldsymbol{u}_{mod} = -\boldsymbol{u}, \theta_{ifmod} = 2\pi - \theta_{if}$. Notice that path planning is executed off-line before the control: in this way, if the planned image trajectories get out camera field of view, it is possible to modify the planning parameters before task execution.

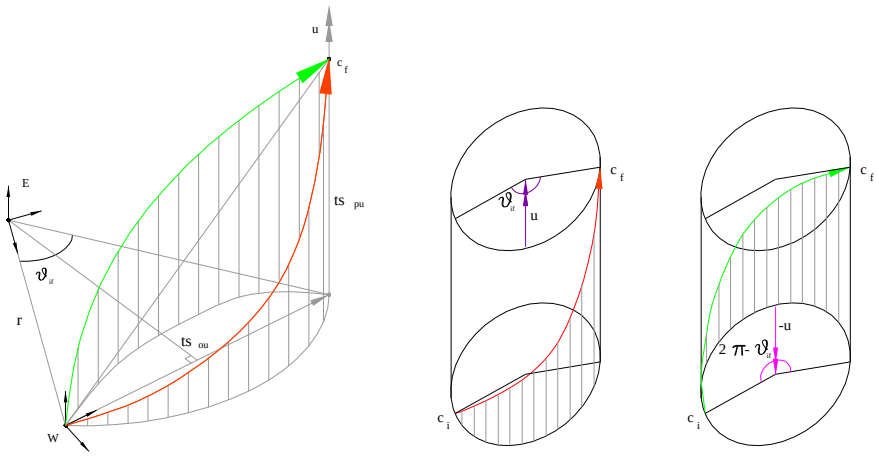


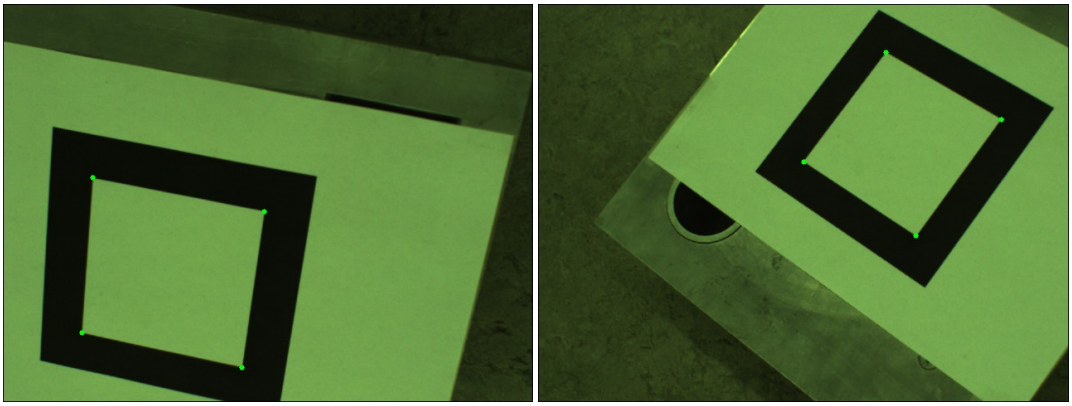
Figure 4: Left: First Modification on helix convexity (plan on the green helix instead of on the red). Right: Second modification (plan on the green helix instead of on the red one).

4 RESULTS

To demonstrate the feasibility of the approach real experiments have been carried out. A 6 DOF Kuka manipulator has been used with a IEEE 1394 firewire camera mounted on its end effector. Control software components has been developed in OROCOS (Open RObot Control Software) a C++ open source tool developed by the Autonomous Compliant Motion Group of the Mechanical Engineering Department in the Katholieke Universiteit of Leuven (Belgium). The control software has been compiled on a real time platform with RTL installed (Real Time Linux), driving the input for the robot actuators. Feature point matching and tracking has been performed by using Intel OpenCV computer vision library. The target was defined by 4 well-detectable points at the vertexes of a 0.09m $\times$ 0.09m square. The visual control (point tracking, $\boldsymbol{e}$ and $\boldsymbol{w}$ computations) has performed at a frequency of 60 Hz (the maximum frame-rate available from the camera) while the robot actuators controller run at frequency of 100 Hz (faster inner feedback control-loop). Firewire camera has been calibrated using the Matlab® Camera Calibration toolbox developed by Jean Yves Bouguet from California Institute of Technology. The resulting intrinsic camera parameters are the following:

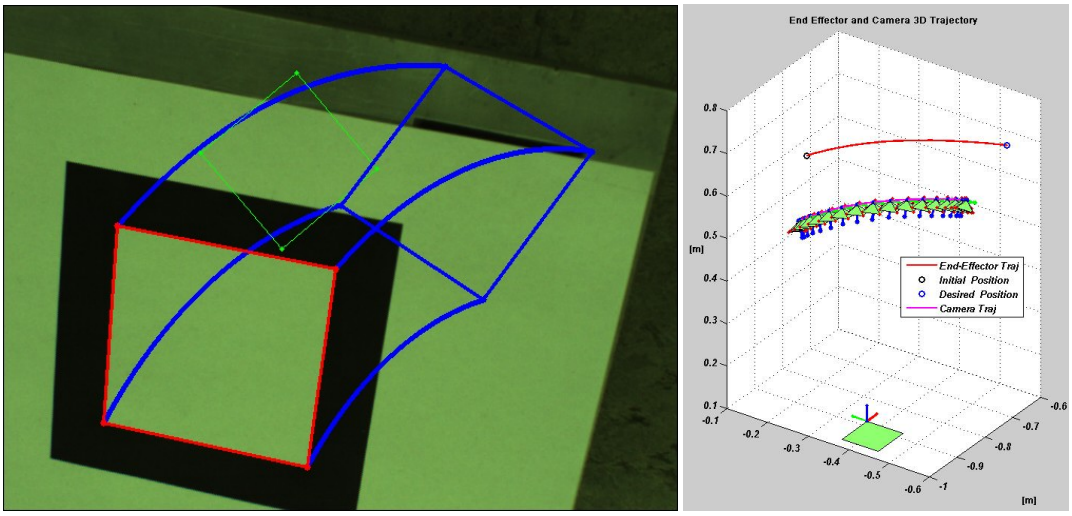
$$\boldsymbol{K} = \begin{bmatrix} 1120.10 & 0 & 342.05 \\ 0 & 1120.19 & 199.57 \\ 0 & 0 & 1 \end{bmatrix}, \tag{32}$$

with a CCD of resolution 640×480. The end effector pose E are defined with respect to the ground frame of the manipulator O by the vector (t_x, t_y, t_z), expressed in [m], identifying E origin with respect to O and by the (RPY) Roll-Pitch-Yaw angles (ϕ, θ, ψ), expressed in $[rad]$, identifying E orientation with respect to O. In the shown experiment the ground truth end-effector poses corresponding respectively to camera poses $< C_i >$ and $< C_f >$ are defined by $(t_{xi} = -0.8, t_{yi} = -0.16, t_{zi} = 0.62)$, $(\phi_i = -1.97, \theta_i = 0.52, \psi_i = 0)$ and by $(t_{xf} = -1.2, t_{yf} = 0.6, t_{zf} = 0.8)$,$(\phi_f = -0.65, \theta_f = -0.55, \psi_f = 0.7)$.



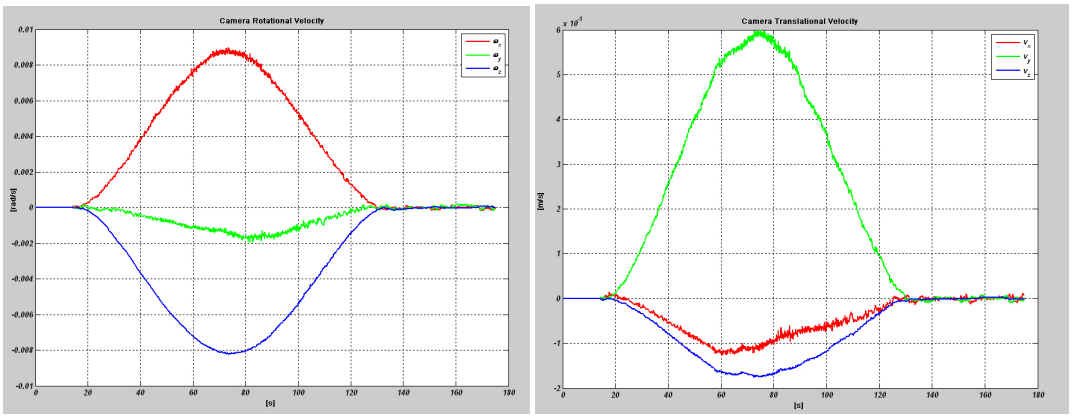
a) b)

Figure 5: a) Initial target view I_i b) Goal target view I_f.



a) b)

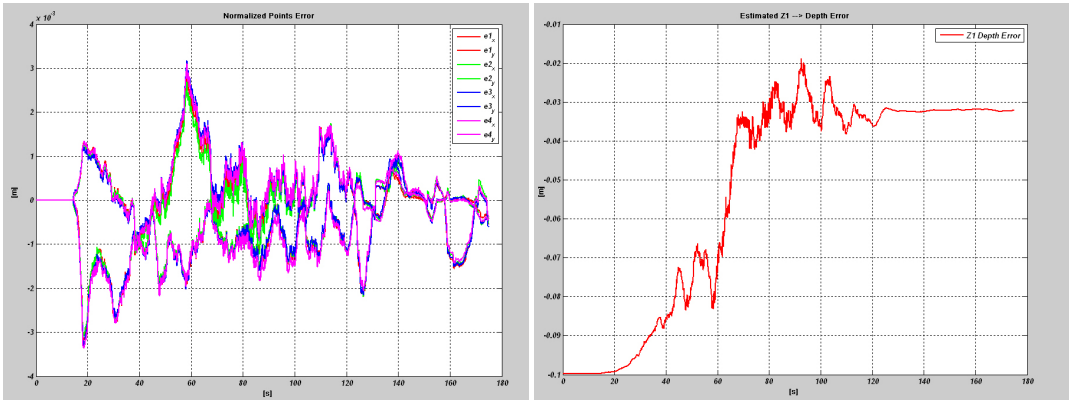
Figure 6: a) Planned image trajectories x_d (blue lines)in the pixel image during the off-line phase, b) 3D camera (magenta line) and end-effector (red line) origins trajectories resulting from the experiments.



a) b)

Figure 7: a) angular camera velocity components $[rad/s]$ b) translational camera velocity components $[m/s]$.

Figure (5) shows the initial and final target views (I_i and I_f) grabbed from the camera; Figure (6) reports the reference image planned trajectory x_d in pixel (blue lines); this figure reports also in thin green the auxiliary view I_a, used tho solve the Euclidean homography decomposition mentioned in Section (2.2). Figure (7) shows the components of the camera twist screw $\boldsymbol{w}$induced from the control law: they result smooth and feasible as set by the quintic polynomial time in equation (24). In Figure



a) b)

Figure 8: a) Normalized image error $\boldsymbol{e}$ components, b) Depth estimation error $[m]$ on the first target point.

(8) a) and b) are respectively reported image error $\boldsymbol{e}$ components and the $3D$ depth estimation error for the first tracked feature-point. As clearly visible $\boldsymbol{e}$ components remain very small during task execution conveniently exploiting the local stability properties of the control law. The adaptive estimation law giving, as shown, has a noisy behavior but converges to a better estimation of the target point unknown depth. 3D camera and end-effector trajectories obtained during task execution

are reported in Figure (6 b): as clearly visible the camera moves on a helical-shape trajectory induced from the image path planning x_d finally resulting in a harmonious movement. The position error $\boldsymbol{e}_{pos}$ reported in $[mm]$ and the orientation error $\boldsymbol{e}_{or}$ reported in $[deg]$ between the reached and the goal one poses are: $\boldsymbol{e}_{pos} = [0.4210\ 0.1240\ -0.3610]^T$ and $\boldsymbol{e}_{or} = [-0.0968\ 0.0162\ -0.0928]^T$ resulting in a very high accuracy of the proposed approach.

5 CONCLUSION

This paper shows a feasible and innovative technique to execute path planning for IBVS positioning task for an eye-in-hand robotic system. Experimental results show the good performances of the control by an image planning scheme resulting in smooth twist-camera screw and high positioning accuracy. Image path planning allows to avoid singularity and local minima problem (drawbacks of classical IBVS approaches) and makes the user able to choose the $3D$ shape of the camera trajectory linking the initial andthe goal $3D$ pose. Future work will address the use of $3D$ more complex targets for task execution.

6 ACKNOWLEDGMENT

Authors would thanks Prof. Joris De Schutter, Prof. Herman Bruyninckx and all the Autonomous Compliant Motion Group from the Katholieke Universiteit of Leuven (Belgium): they made possible to realize real experiments in their lab in a very constructive and friendly environment. A Special thank also to Eng. Ruben Smits and Eng. Kasper Claes for their precious help in control code implementation and in the experimental set-up organization.

References

[Chaumette, 1998] Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In Springer, editor, *The confluence of vision and control (D. Kriegman, G. Hager and A. Morse, Eds.). Lecture Notes in Control and Information Systems*, volume 237, pages 66–78, Barcelona, Spain. on CD-ROM.

[Conticelli and Allotta, 2001] Conticelli, F. and Allotta, B. (2001). Discrete-time robot visual feedback in 3D positioning tasks with depth adaptation. *IEEE/ASME Transaction on Mechatronics*, 6(3):356–363.

[Espiau, 1993] Espiau, B. (1993). Effect of calibration errors on visual servoing in robotics. In *Proc. 3rd International Workshop on Experimental Robotics*, pages 182–192, Kyoto, Japan.

[Espiau et al., 1992] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. Robot. Automat.*, 8(3):313–326.

[Hartley and Zisserman, 2003] Hartley, R. I. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[Hutchinson et al., 1996] Hutchinson, S., Hager, G., and Corke, P. (1996). Tutorial on visual servo control. *IEEE Trans. Robot.*, 12(5):651–670.

[Ma et al., 2003] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. (2003). *An Invitation to 3D Vision: From Images to Geometric Models*. Springer Verlag.

[Malis et al., 1999] Malis, E., Chaumette, F., and Boudet, S. (1999). 2-1/2-d visual servoing. *IEEE Trans. Robot. Automat.*, 15(2):238–250.

Robot Positioning via Image Based Visual Servoing

Duccio Fioravanti[1] , Benedetto Allotta[2] , Rindi Andrea[3]
Department of Energetics, University of Florence, Italy
E-mail: [1] *fioravanti@mapp1.de.unifi.it,* [2] *benedetto.allotta@unifi.it,* [3] *andrea.rindi@unifi.it*

Keywords: Image Based Control, Path Planning, Depth Estimation.

Nowadays the choice of vision sensors to control in real-time complex robotic systems has become a common use, thanks to the increasing performances of both modern computers and hardware components. Various tecniques have been proposed to control the behaviour of a dynamic system exploiting in a feedback control-loop the visual information provided by one ore more cameras: these approaches are known in literature as Visual Servoing [Hutchinson et al., 1996].
Using Visual Servoing a robot can be positioned with respect to a target placed in its workspace, by exploiting the differences between the actual target view (correspondent to an actual robot configuration) and the desired camera view (correspondent to the desired robot configuration). Visually controlled systems may differ in the control architecture (direct and indirect visual servoing), in the number of camera used (mono, stereo or multicamera configurations) and in their placing in the system (eye-in-hand systems or eye-to-hand systems). In this paper the autors will refer to a mono camera system with eye-in-hand configuration (the camera is mounted on the robot end-effector).
Concerning the features of the control scheme, Visual Servoing approaches can be subdivided into three different groups, depending on the definition of the error function adopted to regulate the system: Position Based Visual Servoing (PBVS), Image based Visual Servoing (IBVS) and Hybrid Visual Servoing.
In Position Based Visual Servoing the error is computed by estimating from the images some 3D features of interest (like the desired camera rotation and translation etc...): this approach allows the user to fully define the robot trajectory in 3D space but has proven to be very sensitive with respect to both camera and robot calibration errors and usually requires the knowledge of the $3D$ target model [Hutchinson et al., 1996].
In Image based Visual Servoing the feedback control looop is directly closed in the image since the system error is defined by primitives directly extracted from the images (such as image points, lines, or more complex shapes [Espiau et al., 1992]).
Image based approach in general doesn't require a priori knowledge of the target (model-free approach) and is known to be really robust with respect to both camera and robot calibration errors [Espiau, 1993]. On the other hand only local stability of the controlled system has been proved: that is why, during task exectuions, singularities and local minima of the control law may occur especially when the initial and the desired view of the target are noticeably different (large camera displacements)[Chaumette, 1998]. To overcome singularity and local minima problem a convenient path planning in the image can be done as shown in Figure (1) (see [Allotta and Fioravanti, 2005], [Mezouar and Chaumette, 2002]).
The hybrid visual servoing scheme uses an error defined on both image and cartesian primitives: this method ensures better analitical stability properties (sufficient conditions for global stability has been proved) [Malis et al., 1999]; however also in this approaches 3D robot trajectory is not predictable and the target may go out of the camera field of view, causing the loss of the visual informations (the visual features) and thus the failure of the task.
In this paper authors report experimental results regarding the execution of positioning tasks with

respect to a fixed planar target using a 6 DOF manipulator with a single camera mounted on its end-effector. Positiong task are fulfilled, by the generating an opportune image path-planning so that the robot end-effector is driven to the goal pose following an helicoidal trajectory in 3-D space. The image planning is generated so as to obtain feasible camera velocity-screws compatible with the robot actuators as well.

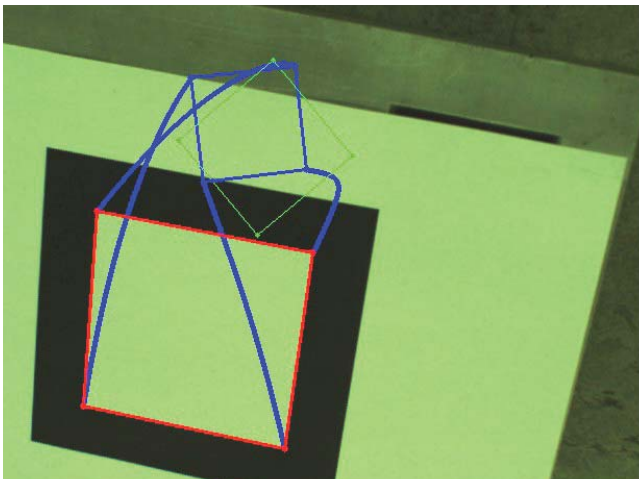


Figure 1: Path planning in the image (blue-lines) for an IBVS task between the initial camera view (red square) and the desired one (blue square) of a planar target.

References

[Allotta and Fioravanti, 2005] Allotta, B. and Fioravanti, D. (2005). 3D motion planning for image-based visual servoing tasks. In *Proc. IEEE Conference on Robotics and Automation*, Barcelona, Spain. on CD-ROM.

[Chaumette, 1998] Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In Springer, editor, *The confluence of vision and control (D. Kriegman, G. Hager and A. Morse, Eds.). Lecture Notes in Control and Information Systems*, volume 237, pages 66–78, Barcelona, Spain. on CD-ROM.

[Espiau, 1993] Espiau, B. (1993). Effect of calibration errors on visual servoing in robotics. In *Proc. 3rd International Workshop on Experimental Robotics*, pages 182–192, Kyoto, Japan.

[Espiau et al., 1992] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. Robot. Automat.*, 8(3):313–326.

[Hutchinson et al., 1996] Hutchinson, S., Hager, G., and Corke, P. (1996). Tutorial on visual servo control. *IEEE Trans. Robot.*, 12(5):651–670.

[Malis et al., 1999] Malis, E., Chaumette, F., and Boudet, S. (1999). 2-1/2-d visual servoing. *IEEE Trans. Robot. Automat.*, 15(2):238–250.

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