Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

3D Motion Planning for Image-Based Visual Servoing Tasks

Benedetto Allotta and Duccio Fioravanti
Dipartimento di Energetica "Sergio Stecco"
University of Florence
Via Santa Marta 3, Firenze, Italy
allotta@unifi.it

***Abstract*— The execution of positioning tasks by using image-based visual servoing can be easier if a trajectory planning mechanism exists. This paper deals with the problem of generating image plane trajectories (a trajectory is made of a path plus a time law) for tracked points in an eye-in-hand system which has to be positioned with respect to a fixed object. The generated image plane paths must be *feasible* i.e. they must be compliant with rigid body motion of the camera with respect to the object so as to avoid image jacobian singularities and local minima problems. In addition, the image plane trajectories must generate camera velocity screws which are smooth and within the allowed bounds of the robot. We show that a scaled 3D motion planning algorithm can be devised in order to generate feasible image plane trajectories.**

***Index Terms*— Image-Based Visual Servoing, Trajectory Planning**

I. INTRODUCTION

Various visual servoing techniques have been proposed to set up control schemes that allow the execution of position tasks by using visual information [1]. In Image-Based Visual Servoing (IBVS) the control loop is closed in the image space and cartesian motion is generated by using the image jacobian (some authors use the term "interaction matrix" in lieu of jacobian matrix). IBVS has received more attention from researchers because it ensures stability and convergence in the presence of modelling [2] and calibration errors [3]. The convergence properties can be improved by using adaptive schemes devoted to estimate on line some of the extrinsic parameters (usually depth) appearing in the image jacobian [4], [5]. However, the execution of a position task by IBVS is not straightforward because singularity problems may occur and local minima may be reached where the reached image is very close to the target one but the 3D positioning task is far from being fulfilled [6]. In [7] a framework is presented to address the problem of reaching a goal image from an initial one while maintaining the object within the camera field of view by means of navigation functions. An alternative approach relies on the generation of image paths for the tracked features, as proposed in [8].

II. DECOMPOSITION OF THE COLLINEATION MATRIX

Consider an initial view I_i and a final or desired view I_f of an unknown fixed object identified by four feature points assumed to be coplanar but not collinear. The images of the feature points in the two views, expressed in homogeneous coordinates, are respectively $\widetilde{\boldsymbol{p}}_j^i = [x_j^i y_j^i 1]^T$ and $\widetilde{\boldsymbol{p}}_j^f = [x_j^f y_j^f 1]^T$, with $j = 1...4$. Let be $\boldsymbol{G}$ the collineation matrix representing the projective homography between the initial and the final images so that

$$\widetilde{\boldsymbol{p}}_j^f = \boldsymbol{G}\widetilde{\boldsymbol{p}}_j^i \quad j = 1...4 \tag{1}$$

From the four given points correspondence we can compute $\boldsymbol{G}$ in several methods [9]. We assume to know $\boldsymbol{K}$, the constant non singular matrix containing the intrinsic camera parameters and so we can find up to a scalar factor the so-called Euclidean homography $\boldsymbol{H} \propto \boldsymbol{K}^{-1}\boldsymbol{G}\boldsymbol{K}$. Using the algorithm expressed in [10], we are able to compute $\boldsymbol{H}$ and to decompose this homography in its rotational ad translational components.
$\boldsymbol{H}$, in fact, can be written as:

$$\boldsymbol{H} = \boldsymbol{R}_{fi} + \frac{\boldsymbol{t}_{fi}}{d_i}\boldsymbol{n}_i^T \quad ; \tag{2}$$

where

- $\boldsymbol{R}_{fi}$ is the rotation matrix between the desired or final camera frame $\boldsymbol{F}_f$ and the initial camera frame $\boldsymbol{F}_i$;
- $\boldsymbol{t}_{fi}$ is the translation between $\boldsymbol{F}_f$ and $\boldsymbol{F}_i$ expressed in $\boldsymbol{F}_f$;
- d_i is the distance from the plane defined by the feature points (denoted by Π) and the origin of $\boldsymbol{F}_i$;
- $\boldsymbol{n}_i$ is the unitary normal to Π expressed in $\boldsymbol{F}_i$.

The algorithm described in [10] gives four different sets of solutions for the variables R_{fi}, $\left(\frac{\boldsymbol{t}_{fi}}{d_i}\right)$ and $\boldsymbol{n}_i$. In order to find the real set from the four given we need an auxiliary view (denoted by I_a), that differs from the other two at least for a translation. So, from the images I_i, I_f,I_a and $\boldsymbol{K}$, we determine the rotation and the translation up to the scalar factor d_i between the camera frames $\boldsymbol{F}_f$ and $\boldsymbol{F}_i$ and we also obtain a scaled Euclidean reconstruction of the unknown object up to the same scalar factor. If we consider that

$$\boldsymbol{R}_{if}{}^T = \boldsymbol{R}_{fi} \qquad \boldsymbol{t}_{fi} = -\boldsymbol{R}_{fi}\boldsymbol{t}_{if} \tag{3}$$

we can rewrite equation (2) as:

$$\boldsymbol{H} = \boldsymbol{R}_{if}{}^T(\boldsymbol{I} - \frac{\boldsymbol{t}_{if}}{d_i}\boldsymbol{n}_i{}^T) \quad , \tag{4}$$

where all elements are expressed respect the initial camera frame $\boldsymbol{F}_i$.

The collineation matrix can be finally rewritten as:

$$\boldsymbol{G} = \boldsymbol{K}\boldsymbol{R}_{if}{}^T(\boldsymbol{I} - \frac{\boldsymbol{t}_{if}}{d_i}\boldsymbol{n}_i{}^T)\boldsymbol{K}^{-1} \quad . \tag{5}$$

III. PLANNING OF THE DESIRED CAMERA TRAJECTORY

To improve the performance of our visual servoing control we build a sequence of reference collineation matrices that makes the reference image gradually vary from the initial image I_i at time $t = t_i = 0$ to the final desired image I_f at time $t = t_f$, considering also the compatibility of this sequence with the rigid motion of the camera. This sequence is represented by the parameter-dependent matrix:

$$\boldsymbol{G}_d(s) = \boldsymbol{K}\boldsymbol{H}_d(s)\boldsymbol{K}^{-1} \quad , \tag{6}$$

where $s = s(\tau)$ is a mononotonic function of the non dimensional time $\tau = \frac{t}{t_f}$ ranging from $s_0 = s(0)$, corresponding to $t = 0$ to $s_f = s(1)$, corresponding to $t = t_f$. $\boldsymbol{H}_d(s)$ must verify the following conditions:

$$\boldsymbol{H}_d(s_0) = \boldsymbol{I} \qquad \boldsymbol{H}_d(s_f) = \boldsymbol{H} \quad .$$

The parameter $s(\tau)$ may be, for instance, a function of the form:

$$s(\tau) = a\tau^5 + b\tau^4 + c\tau^3 + d\tau^2 + e\tau + f \quad , \tag{7}$$

In order to have smooth variations of $\boldsymbol{H}_d(s)$, $s(\tau)$ must verify the following conditions:

$$\begin{array}{lcll} s(0) = 0 & \dot{s}(0) = 0 & \ddot{s}(0) = 0 & \\ s(1) = 1 & \dot{s}(1) = 0 & \ddot{s}(1) = 0 & . \end{array} \tag{8}$$

From (7) and (8) we get:

$$a = 6 \qquad b = -15 \quad c = 10 \quad d = e = f = 0 \quad . \tag{9}$$

Considering the decomposition (4) we can have four different cases:

- $\boldsymbol{R}_{if} \neq \boldsymbol{I}$ and $\frac{\boldsymbol{t}_{if}}{d_i} \neq \boldsymbol{0}$; this is the most general case;
- $\boldsymbol{R}_{if} \neq \boldsymbol{I}$ and $\frac{\boldsymbol{t}_{if}}{d_i} = \boldsymbol{0}$; pure rotational motion case;
- $\boldsymbol{R}_{if} = \boldsymbol{I}$ and $\frac{\boldsymbol{t}_{if}}{d_i} \neq \boldsymbol{0}$; pure translational motion case;
- $\boldsymbol{R}_{if} = \boldsymbol{I}$ and $\frac{\boldsymbol{t}_{if}}{d_i} = \boldsymbol{0}$; trivial case (the camera is already in the desired frame).

We will first analyze the general motion case and then the other two cases of interest.

A. General motion case

In this case we have to produce both a rotation and a translation of the camera to reach the desired view I_f, so we can write the time dependent reference Euclidean homography $\boldsymbol{H}_d(s)$ as follow:

$$\boldsymbol{H}_d(s) = \boldsymbol{R}_d(s)^T(\boldsymbol{I} - \frac{\boldsymbol{t}_d}{d_i}(s)\boldsymbol{n}_i{}^T) \quad ; \tag{10}$$

with

$$\boldsymbol{R}_d(0) = \boldsymbol{I}, \quad \boldsymbol{R}_d(1) = \boldsymbol{R}_{if}, \quad \frac{\boldsymbol{t}_d}{d_i}(0) = \boldsymbol{0}, \quad \frac{\boldsymbol{t}_d}{d_i}(1) = \frac{\boldsymbol{t}_{if}}{d_i}.$$

The reference rotation matrix $\boldsymbol{R}_d(s)$ can be expressed through the exponential form:

$$\boldsymbol{R}_d(s) = \boldsymbol{I} + [\boldsymbol{u}]\sin(\theta_d(s)) + [\boldsymbol{u}]^2(1 - \cos(\theta_d(s)) \quad ; \tag{11}$$

assuming the reference angle of rotation $\theta_d(s) = \theta_{if}s$ we finally have:

$$\boldsymbol{R}_d(s) = \boldsymbol{I} + [\boldsymbol{u}]\sin(\theta_{if}s) + [\boldsymbol{u}]^2(1 - \cos(\theta_{if}s)) \quad ; \tag{12}$$

where the set $(\boldsymbol{u}, \theta_{if})$ is the axis-angle representation of the matrix $\boldsymbol{R}_{if}$. We have:

$$\begin{aligned} \cos(\theta_{if}) &= \frac{1}{2}(tr(\boldsymbol{R}_{if}) - 1), \\ \sin(\theta_{if}) &= \sqrt{1 - \cos(\theta_{if})^2}, \\ \theta_{if} &= \mathrm{atan2}(\sin(\theta_{\mathrm{if}}), \cos(\theta_{\mathrm{if}})), \end{aligned} \tag{13}$$

and

$$[\boldsymbol{u}] = \frac{\boldsymbol{R}_{if}{}^T - \boldsymbol{R}_{if}}{2\sin(\theta_{if})} \quad ; \tag{14}$$

with

$$[\boldsymbol{u}] = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \quad .$$

if $\theta_{if} = \pi$, $\boldsymbol{u}$ is the unit length eigenvector or $\boldsymbol{R}_{if}$ associated with the eigenvalue 1. From expression (13) we always obtain $0 \leq \theta_{if} \leq \pi$. To define the time dependent reference scaled translation of the camera $\frac{\boldsymbol{t}_d}{d_i}(s)$ we must choose a trajectory that identifies the position of the reference camera frame origin during the task.

Since the parameter d_i is unknown, we only are able to define the shape of the reference trajectory and not its real dimension (we define, in fact, a scaled trajectory). Notice that whatever the unknown scale factor, the desired trajectories in the image plane remain the same. So, if the control is synthesized in the image plane, the positioning task can be fulfilled regardless of the scale factor. Moreover we observe that the knowledge of d_i is equivalent to the knowledge of the real dimensions of the whole scene (object of interest and reference trajectory).

Let be $\boldsymbol{ts}_{if} = \frac{\boldsymbol{t}_{if}}{d_i}$ the scaled desired translation vector and $\boldsymbol{ts}_d = \frac{\boldsymbol{t}_d}{d_i}(s)$ the time dependent reference scaled translation vector. Among the infinite shapes for the reference translation we choose for our planning the helicoidal trajectory; we consider the helicoidal shape as the most appropriate to join the initial and desired positions of the camera since it perfectly harmonizes traslation with rotation. A generic helix is characterized by these parametric equations:

$$\left\{ \begin{array}{l} x(\gamma(s)) = r\cos(\gamma(s)) \\ y(\gamma(s)) = r\sin(\gamma(s)) \\ z(\gamma(s)) = \frac{p}{2\pi}\gamma(s) \end{array} \right. \tag{15}$$

$$0 \leq \gamma(s) \leq \gamma_1 \quad ,$$

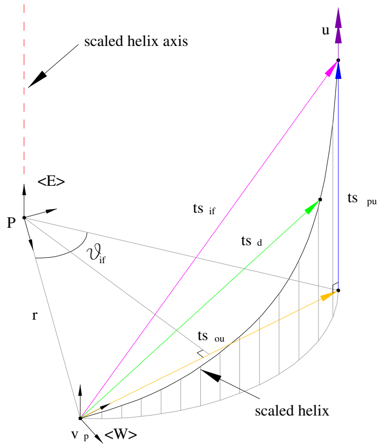

Fig. 1. Reference scaled helix used for the camera translation path in the general motion case.

where x,y and z are the coordinates of the curve expressed in a canonical cartesian frame defined by E. This right-handed coordinate system has the z-axis coincident with the axis of the helix and the x-axis defined by the vector perpendicular to the helix axis through initial point of the curve (the point defined by $s = 0$). The helix we choose has the z-axis direction coincident with the unit vector $\boldsymbol{u}$ and the total angle γ_1 covered by the curve equal to θ_{if}, as shown in Fig. 1. We also assume :

$$\gamma(s) = \theta_d(s) = \theta_{if} s \quad ; \tag{16}$$

in this way we obtain for the camera motion the translation perfectly harmonized with the rotation. To completely define our scaled reference helix we must still find:

- the pitch of the helix p ;
- the radius of the helix r ;
- the origin of the canonical coordinate system E in the 3D space.

In order to find these parameters we decompose the vector $\boldsymbol{ts}_{if}$ in its parallel and orthogonal components to the unit vector $\boldsymbol{u}$, respectively $\boldsymbol{ts}_{pu}$ and $\boldsymbol{ts}_{ou}$. We have:

$$\boldsymbol{ts}_{pu} = (\boldsymbol{ts}_{if} \bullet \boldsymbol{u})\boldsymbol{u} \quad , \tag{17}$$

$$\boldsymbol{ts}_{ou} = \boldsymbol{ts}_{if} - \boldsymbol{ts}_{pu} \quad . \tag{18}$$

Now we have another three sub cases:

(1) - $\boldsymbol{ts}_{pu} \neq \boldsymbol{0}$ and $\boldsymbol{ts}_{ou} \neq \boldsymbol{0}$;
(2) - $\boldsymbol{ts}_{pu} \neq \boldsymbol{0}$ and $\boldsymbol{ts}_{ou} = \boldsymbol{0}$;
(3) - $\boldsymbol{ts}_{pu} = \boldsymbol{0}$ and $\boldsymbol{ts}_{ou} \neq \boldsymbol{0}$.

1) Sub case: $\boldsymbol{ts}_{pu} \neq \boldsymbol{0}$ and $\boldsymbol{ts}_{ou} \neq \boldsymbol{0}$: In this sub case it's possible to verify the following relation:

$$p/(2\pi) = (\boldsymbol{ts}_{if}{}^T \bullet \boldsymbol{u})/\theta_{if} \quad \Rightarrow \quad p = \frac{2\pi}{\theta_{if}}(\boldsymbol{ts}_{if}{}^T \bullet \boldsymbol{u}) \quad ; \tag{19}$$

we can note that if we have $\boldsymbol{ts}_{pu} \bullet \boldsymbol{u} > 0$ we have a *right* helix while if $\boldsymbol{ts}_{pu} \bullet \boldsymbol{u} < 0$ we have a *left* helix. With simple geometric passages we obtain the radius as:

$$r = \frac{|\boldsymbol{ts}_{ou}|}{2\sin(\theta_{if}/2)} \quad . \tag{20}$$

The radius and the pitch of the real reference helix, respectively r_r and p_r, are unknown variables expressed from these relations:

$$r_r = r d_i \quad , \quad p_r = p d_i \quad . \tag{21}$$

To fix the scaled helix in the space we have to finally find the coordinate of the origin of the frame E, defined as P, respect the initial frame F_i of the camera.
Let be W a right-handed coordinate system with the same origin of F_i and orthogonal unit vectors defined as follow:

$$\begin{cases} \boldsymbol{v}_p &= \frac{\boldsymbol{ts}_{ou}}{|\boldsymbol{ts}_{ou}|} \wedge \boldsymbol{u} \\ \boldsymbol{v}_{t\perp u} &= \frac{\boldsymbol{ts}_{ou}}{|\boldsymbol{ts}_{ou}|} \\ \boldsymbol{u} & \end{cases} \quad ; \tag{22}$$

from the last relations we can compute the 4×4 transformation matrix from F_i to W:

$$\boldsymbol{M}_W^{F_i} = \begin{bmatrix} \boldsymbol{v}_p & \boldsymbol{v}_{t\perp u} & \boldsymbol{u} & \boldsymbol{0} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad . \tag{23}$$

The position and the orientation of E expressed in W are respectively represented by the vector $\boldsymbol{ts}_E^W$ and by the rotation matrix $\boldsymbol{R}_E^W$; from geometry, we have:

$$\boldsymbol{ts}_E^W = [\ -r\cos(\theta_{if}/2) \quad r\sin(\theta_{if}/2) \quad 0\ ]^T \quad , \tag{24}$$

$$\boldsymbol{R}_E^W = \begin{bmatrix} \cos(\theta_{if}/2) & \sin(\theta_{if}/2) & 0 \\ -\sin(\theta_{if}/2) & \cos(\theta_{if}/2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad . \tag{25}$$

The transformation matrix from W to E can be expressed as:

$$\boldsymbol{M}_E^W = \begin{bmatrix} \boldsymbol{R}_e^w & \boldsymbol{ts}_E^W \\ \boldsymbol{0}^T & 1 \end{bmatrix} \quad . \tag{26}$$

Now we are able to identify the placement of E from the coordinate system F_i through the transformation matrix $\boldsymbol{M}_E^{F_i}$:

$$\boldsymbol{M}_E^{F_i} = \boldsymbol{M}_W^{F_i}\boldsymbol{M}_E^W \quad . \tag{27}$$

From the equations (15) and (16) we obtain the reference scaled position of the camera expressed in E:

$$\boldsymbol{ts}_{d(E)} = \begin{cases} x(s) = r\cos(\theta_{if}s)) \\ y(s) = r\sin(\theta_{if}s) \\ z(s) = \frac{p}{2\pi}\theta_{if}s \end{cases} \tag{28}$$

Using equation (27) we obtain the reference scaled translation expressed in F_i:

$$\tilde{\boldsymbol{ts}}_d = \boldsymbol{M}_E^{F_i}\tilde{\boldsymbol{ts}}_{d(E)}, \tag{29}$$

with $\tilde{\boldsymbol{ts}}_d = [\boldsymbol{ts}_d, 1]^T$ and $\tilde{\boldsymbol{ts}}_{d(E)} = [\boldsymbol{ts}_{d(E)}, 1]^T$.
Now we are able compute the reference Euclidean homograpy and the reference collineation matrix using respectively equations (10) and (6). To find the reference

trajectories in the image space we finally use this equation:

$$\widetilde{\boldsymbol{p}}_j^d = \begin{bmatrix} x_j^d \\ y_j^d \\ 1 \end{bmatrix} = \boldsymbol{G}_d(s) \begin{bmatrix} x_j^i \\ y_j^i \\ 1 \end{bmatrix} = \boldsymbol{G}_d(s)\widetilde{\boldsymbol{p}}_j^i \quad j = 1...4 \tag{30}$$

where $\widetilde{\boldsymbol{p}}_j^d$ is the generic reference image point expressed in homogeneus coordinates.

2) Sub case: $\boldsymbol{ts}_{pu} \neq \boldsymbol{0}$ and $\boldsymbol{ts}_{ou} = \boldsymbol{0}$: From equation (20) we have in this sub case $r = 0$. In this situation the helix degenerates into a line and the scaled reference translation can easily be computed:

$$\boldsymbol{ts}_d = s\boldsymbol{ts}_{if} \tag{31}$$

3) Sub case: $\boldsymbol{ts}_{pu} = \boldsymbol{0}$ and $\boldsymbol{ts}_{ou} \neq \boldsymbol{0}$: From the equations (17) and (19) we have $p = 0$. In this sub case the helix degenerates into an arc of circle and the vector $\boldsymbol{ts}_{d(E)}$ of equation (28) become:

$$\boldsymbol{ts}_{d(E)} = \begin{cases} x(s) = r\cos(\theta_{if}s)) \\ y(s) = r\sin(\theta_{if}s) \\ z(s) = 0 \end{cases} . \tag{32}$$

B. Pure rotational motion case

In this situation we have only to produce a rotation of the camera to reach the desired view, in fact equation (4) become:

$$\boldsymbol{H} = \boldsymbol{R}_{if}{}^T \quad . \tag{33}$$

We extract as in in the previous subsection the unit vector $\boldsymbol{u}$ and the angle θ_{if} from the rotation matrix $\boldsymbol{R}_{if}$ and we plan the rotation of the camera using equation (12):

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

C. Pure translational motion case

In this last case equation (4) become:

$$\boldsymbol{H} = (\boldsymbol{I} - \frac{\boldsymbol{t}_{if}}{d_i}\boldsymbol{n}_i{}^T) \tag{34}$$

and the camera need only to translate to reach the desired position. We plan the desired trajectory along a line joining the initial and desired position using the parameterization expressed in (31):

$$\boldsymbol{ts}_d = s\boldsymbol{ts}_{if}$$

IV. Optimization of the desired trajectory

The choice of use an helicoidal planning for the reference trajectory of the camera needs some further modifications to be optimized for any possible initial and desired view of the target.

The first modification comes from the fact that the reference helicoidal trajectory extends in the robot workspace and in some cases, because of its convexity, brings the camera too much close to the target: in this situation some feature point can leave the camera field of view and the task is compromised. To avoid these cases it is sufficient *to have the convexity of the helicoidal reference trajectory always turned from the opposite part respect to the target.* To reach this goal we at first consider the frame W defined in (22): we note that any helix produced from the planning lies in the half space delimited by the coordinate plane with normal $\boldsymbol{v}_p$ characterized by its positive direction. Since we assume the normal $\boldsymbol{n}_i$ to the target plane Π, obtained from the decomposition of $\boldsymbol{H}$, to have its positive direction towards the half space delimited by Π containing the target, we operate in this manner:

- we maintain the helix generated from the previous planning if $\boldsymbol{v}_p \bullet \boldsymbol{n}_i \leq 0$
- we instead modify the reference helix if $\boldsymbol{v}_p \bullet \boldsymbol{n}_i > 0$ (in these cases, in fact, the convexity of the curve is turned towards the target).

The change we operate if $\boldsymbol{v}_p \bullet \boldsymbol{n}_i > 0$ is to plan the reference position trajectory on an helix symmetric to the old one respect to the coordinate plane of W with normal $\boldsymbol{v}_p$. To operate this symmetry it is sufficient to introduce a matrix of the form:

$$\boldsymbol{S} = \text{diag}(-1, 1, 1, 1) \quad . \tag{35}$$

in equation (27). This equation become:

$$\boldsymbol{M}_{E_{simm}}^{F_i} = \boldsymbol{M}_W^{F_i}\boldsymbol{S}\boldsymbol{M}_E^W \quad , \tag{36}$$

where with E_{simm} we have defined the canonical frame associated to the modified symmetric helix. The second modification we produce to optimize the camera translation path rises from the fact that in some complex tasks the orientation planning brings the camera to turn towards the opposite part respect to the target: also in these cases the target gets out of the camera field of view and the task is compromised. These unfortunate cases might arise when the initial view, the desired or both views are particular views in which the camera focal axis intersects the object plane Π in a point with negative depth. To resolve this problem we plan the camera orientation and translation on a different helix characterized by these new parameters:

$$\boldsymbol{u}_{new} = -\boldsymbol{u} \quad , \quad \theta_{ifnew} = 2\pi - \theta_{if} \quad . \tag{37}$$

This modification makes the camera to join the desired frame always maintaining the camera towards the target. Notice that, even if in some unfortunate cases the feature points would leave the camera field of view, the paths planning in the image space permits to prevent these situations and so to avoid them choosing a different reference scaled 3D trajectory for the camera.

V. Adaptive Visual Servoing

In this section we present the visual servoing control used to simulate the tasks. Let be

$$\boldsymbol{x} = [\ \boldsymbol{p}_1^T \quad \boldsymbol{p}_2^T \quad \boldsymbol{p}_3^T \quad \boldsymbol{p}_4^T\ ]^T \quad ,$$
$$\boldsymbol{x}^d = [\ (\boldsymbol{p}_1^d)^T \quad (\boldsymbol{p}_2^d)^T \quad (\boldsymbol{p}_3^d)^T \quad (\boldsymbol{p}_4^d)^T\ ]^T$$

the vectors $\in \boldsymbol{R}^{8\times1}$ containing respectively the image of the four considered feature points and the coordinate of the correspondent reference feature points (given by the planning defined in the previous sections) expressed in non homogeneous coordinate.
The system dynamics is:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\boldsymbol{w} \quad , \tag{38}$$

where $\boldsymbol{w} = [\ \boldsymbol{v}^T \quad \boldsymbol{\omega}^T\ ]^T \in \boldsymbol{R}^{6\times1}$ is the twist velocity screw of the camera expressed in the camera frame and $\boldsymbol{J} = [\ \boldsymbol{J}_1^T \quad \boldsymbol{J}_2^T \quad \boldsymbol{J}_3^T \quad \boldsymbol{J}_4^T\ ]^T$ is the so called interaction matrix. Using for the camera the basic pinhole model with $\boldsymbol{K} = diag(f, f, 1)$ (where f is the camera focal length) a generic element $\boldsymbol{J}_p$ of $\boldsymbol{J}$ with $p = 1...4$ is a matrix of the form:

$$\begin{bmatrix} -\dfrac{f}{Z_c} & 0 & \dfrac{x_c}{Z_c} & \dfrac{x_c y_c}{f} & -\left(f + \dfrac{x_c^2}{f}\right) & y_c \\ 0 & -\dfrac{f}{Z_c} & \dfrac{y_c}{Z_c} & \left(f + \dfrac{y_c^2}{f}\right) & -\dfrac{x_c y_c}{f} & -x_c \end{bmatrix}$$

where x_c and y_c are the coordinate of a generic image point and Z_c is the depth of the same point respect to the camera frame. At each step of the control we compute the error $\widetilde{\boldsymbol{x}} = \boldsymbol{x}^d - \boldsymbol{x}$ between the reference and the real image of the feature points of the target. The following control law can be used:

$$\boldsymbol{w} = k\hat{\boldsymbol{J}}^{\dagger}\widetilde{\boldsymbol{x}} \quad , \tag{39}$$

where k is a constant gain and $\hat{\boldsymbol{J}}^{\dagger}$ is the pseudo-inverse of the estimated $\boldsymbol{J}$. A proper choice of k can ensure system stability as shown in [2]. To estimate the unknown feature point depth Z_c we use the adaptive law defined in [4].

VI. Simulations

For the two reported simulations we have used a focal length $f = 0.01$m and a square CCD sensor of side $d = 0.01$m. The target object has 4 well detectable points at the vertexes of a 0.4m×0.4m square. The initial and desired camera poses are defined with respect to an object frame having its origin at the center of the square and the z-axis pointing inside the object plane. A pose is hence defined by the origin translation components t_x, t_y, t_z expressed in m and the ZYZ Euler angles ϕ, θ, ψ expressed in deg.

a) Simulation 1: The initial camera pose is: $t_x = -0.6$, $t_y = 0.7$, $t_z = -1$, $\phi = -35$, $\theta = 40$, $\psi = 65$ while the desired camera pose is: $t_x = 1.5$, $t_y = 2.8$, $t_z = -3$, $\phi = 90$, $\theta = -50$, $\psi = 170$. The initial and desired object views are shown in Fig. 2.a and 2.b respectively. The translational and angular velocities resulting from the proposed planning method and the control law (39) are shown in Fig. 2.c and 2.d. Fig. 3.a and 3.b. respectively show the image errors $\widetilde{\boldsymbol{x}}$ and the estimated depth errors of the feature points. Fig. 3.c and 3.d show the trajectories followed by the image points an by the camera in 3D space. As it can be seen, the control law drives the object points along the planned image trajectories, the camera twist screw components are smooth, and the camera pose converges to the target one.

b) Simulation 2: In this simulation we show the execution of a more complex task. The initial camera pose is: $t_x = -0.4$, $t_y = 0.5$, $t_z = -0.1$, $\phi = -25$, $\theta = 30$, $\psi = 35$ while the desired camera pose is: $t_x = 0.6$,$t_y = 1.8$,$t_z = -0.1$, $\phi = 90$, $\theta = -100$, $\psi = 20$. The initial and desired object views are shown in Fig. 4.a and 4.b respectively. The translational and angular velocities resulting from the proposed planning method and the control law (39) are shown in Fig. 4.c and 4.d. In Fig. 5.a and 5.b. are shown respectively the components of the error $\widetilde{\boldsymbol{x}}$ and the errors in the estimated depths of the feature points. In Fig. 5.c and 5.d are shown the trajectories followed by the image points an by the camera in 3D space. Once again the camera twist screw components are smooth and the camera pose converges to the desired one.

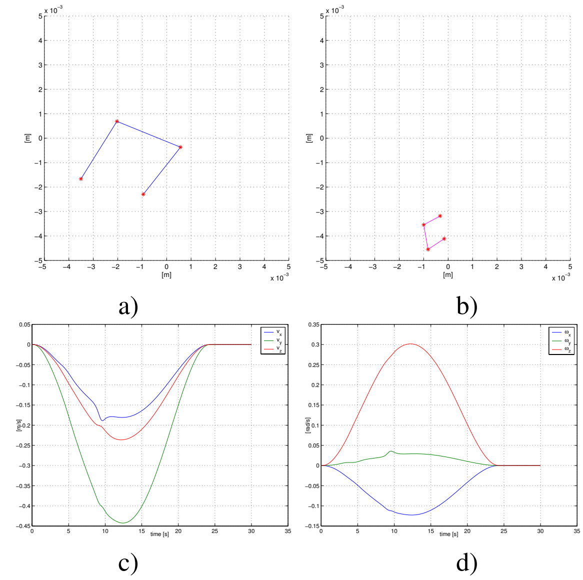

Fig. 2. Simulation 1: a) initial view, b) desired view, c) translational velocity components versus time, d) angular velocity components versus time.

VII. Conclusion

Starting from initial and desired image views of an object, a framework has been proposed to plan feasible image space trajectories for the tracked points. The method includes a first step where a scaled Euclidean reconstruction of the scene is made and a scaled 3D trajectory for the camera is generated. The 3D scaled trajectory is then used to generate a sequence of object views via the homography decomposition. The generated image reference trajectories can be used to set up an IBVS scheme relying on the image jacobian, updated on line by a depth adaptation law. Simulations are reported in order to demonstrate the feasibility of the approach even if the initial and desired object view are very different.

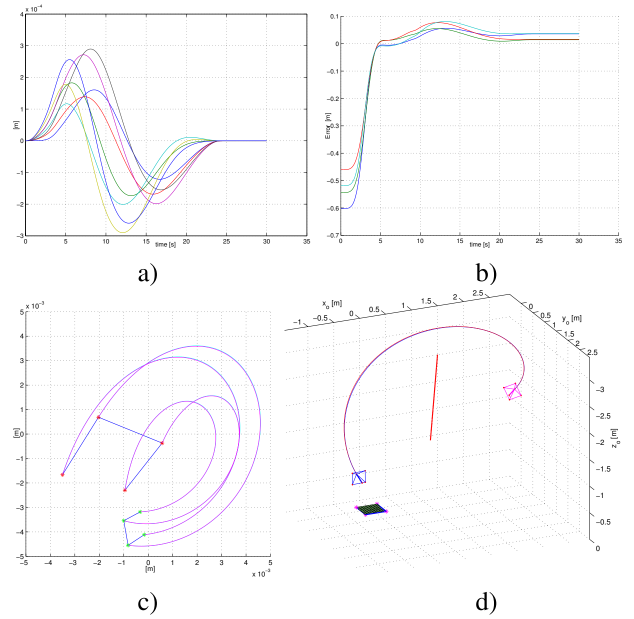

Fig. 3. Simulation 1: a) image errors, b) estimated depth errors, c) desired and real trajectories in the image space, d) reference (scaled) and real camera trajectories in 3D space.

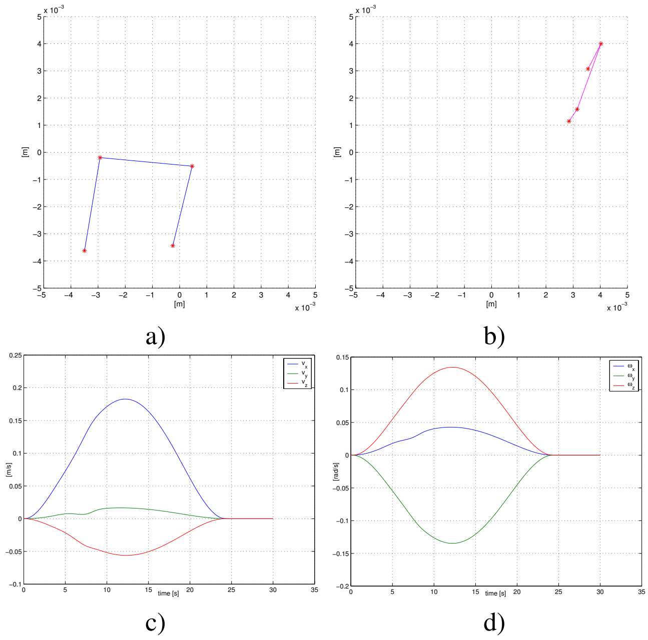

Fig. 4. Simulation 2: a) initial view, b) desired view, c) translational velocity components versus time, d) angular velocity components versus time.

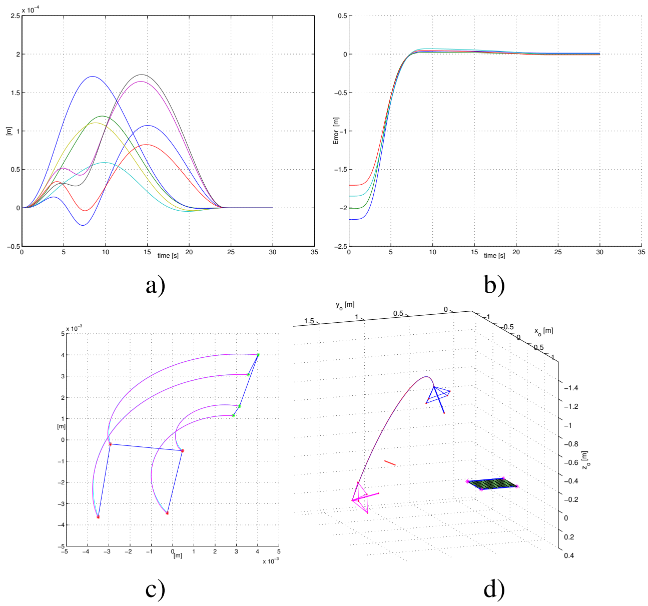

Fig. 5. Simulation 2: a) image errors, b) estimated depth errors, c) desired and real trajectories in the image space, d) reference (scaled) and real camera trajectories in 3D space.

REFERENCES

[1] S. Hutchinson, G.D. Hager, P.I, Corke, "Tutorial on Visual Servo Control," *IEEE Transactions on Robotics and Automation*, Vol. 12 No. 5, 651–670, 1996

[2] B. Espiau, F. Chaumette, P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, 313-326, 1992

[3] B. Espiau, "Effect of calibration errors on visual servoing in robotics," *3rd International Workshop on Experimental Robotics*, Kyoto, Japan, 182-192, October 1993

[4] F. Conticelli, B. Allotta, "Discrete-time robot visual feedback in 3D positioning tasks with depth adaptation,"*IEEE/ASME Transaction on Mechatronics,* Vol. 6, No. 3, 356-363, September 2001.

[5] W. E. Dixon, Y. Fang, D. M. Dawson "Adaptive Range Identification for Exponential Visual Servo Tracking", in *The 18th IEEE International Symposium on Intelligent Control*, ISIC'03, Paper ID: ISIC-0287, 2003.

[6] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing", in *The confluence of vision and control*, D. Kriegman, G. Hager, A. Morse (eds.), Lecture Notes in Control and Information Systems, Vol. 237, pp. 66-78, Springer-Verlag, 1998.

[7] N.J. Cowan, J.D. Weingarten, D.E. Koditschek, "Visual servoing via navigation functions" *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 4, 521-533, August 2002.

[8] Y. Mezouar, F. Chaumette, "Path Planning for Robust Image-Based Control," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 4, 534-549, August 2002.

[9] R. Hartley, A. Zissermann, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2000.

[10] Y. Ma, S. Soatto, J. Kosecka, S. Sastry "An Invitation to 3D Vision:: From Images to Geometric Models ", Springer Verlag, 2003.