



# Image interpolation for image-based control under large displacement

Youcef Mezouar, A. Remazeilles, Patrick Gros, François Chaumette

## ► To cite this version:

Youcef Mezouar, A. Remazeilles, Patrick Gros, François Chaumette. Image interpolation for image-based control under large displacement. IEEE Int. Conf. on Robotics and Automation, ICRA'02, 2002, Washington DC, United States. pp.3787-3794. inria-00352099

**HAL Id: inria-00352099**

**<https://inria.hal.science/inria-00352099>**

Submitted on 12 Jan 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Images Interpolation for Image-based Control Under Large Displacement

Youcef Mezouar, Anthony Remazeilles, Patrick Gros, François Chaumette

IRISA - INRIA Rennes  
Campus de Beaulieu,  
35042 Rennes Cedex, France

## Abstract

*The principal deficiency of image-based visual servoing is that the induced (3D) trajectories are not optimal and sometimes, especially when the displacement to realize is large, these trajectories are not physically valid leading to the failure of the servoing process. Furthermore, visual control needs a matching step between the features extracted from the initial image and the desired one. This step can be problematic (or impossible) when the camera displacement between the acquisitions of the initial and desired images is large and/or for complex scenes. To resolve these deficiencies, we couple an image interpolation process between  $N$  relay images extracted from a database and an image-based trajectory tracking. The camera calibration and the model of the observed scene are not assumed to be known. The relay images are interpolated in such a way that the corresponding camera trajectory is minimal. First a closed form collineation path is obtained and then the analytical form of image features trajectories are derived and efficiently tracked using a purely image-based control. Experimental results obtained on a six DOF eye-in-hand robotic system are presented and confirm the validity of the proposed approach.*

## 1 Introduction

Image-based visual servoing is now a well known control framework [10], [13]. In this approach, the reference image of the object corresponding to a desired position of the robot is generally acquired first (during an off-line step), and some image features are extracted. Features extracted from the initial image are matched with those obtained from the desired one. These features are then tracked during the camera (and/or the object) motion, using for example a correlation based method. An error is obtained by comparing the image features in the current image and in the reference one. The robot motion is then controlled in order to minimize the error (using for example a gradient descent approach). Since the error is directly measured in the image, image-based servo has some degrees of robustness with respect to modeling errors and noise perturbations. However,

sometimes, and especially when the initial and desired configurations are distant, the trajectories induced by image-based servo are neither physically valid nor optimal due to the nonlinearity and potential singularities in the relation from the image space to the workspace [1]. Furthermore, when the camera displacement between the acquisitions of the initial and desired images is large and/or when the observed scene is complex, the matching step can be difficult and even impossible (for example when no joint features could be detected in the considered images).

Dealing with the first problem, path planning in the image-space is a promising approach. Indeed, if the initial error is too large, a reference trajectory can be designed from a sequence of images. The initial error can thus be sampled so that at each iteration of the control loop the error to regulate remains small. In [11], a potential switching control scheme and relay images that interpolate initial and reference image features using an affine approximation are proposed to enlarge the stable region. In [12], a trajectory generator using a stereo system is proposed and applied to obstacle avoidance. An alignment task for a 4 DOF robot using intermediate views of the object synthesized by image morphing is presented in [22]. A path planning for a straight-line robot translation observed by a weakly calibrated stereo system is performed in [19]. In previous work [16], we have proposed a potential field-based path planning generator that determines the trajectories in the image of a set of points lying on an unknown target. To increase the stability region, Cowan and Koditschek describe in [3] a globally stabilizing method using navigation function. However, none of these previous works is dealing with optimality issues. In [23], a numerical framework for the design of optimal trajectories in the image space is described and applied to planar mobile robot with a one dimensional camera. In [17], we give an analytical solution to optimal path planning in the image space (with respect to minimum camera acceleration criterion) between two given images.

Classical visual servoing techniques make the assumptions on the link between the initial image and the desired one. If the initial and desired images are very different

and/or the scene is complex, the steps of finding and matching joint image features (abbreviated SFMJF) are uneasy, and even sometimes impossible if no features belongs to both images. In such a case, the servoing can not be realized. To cope with this deficiency, we propose to use a set of relay images (such that between two successive images, SFMJF are feasible). This set of relay images is automatically extracted from an image database obtained and indexed off-line. When the SFMJ have been realized between each pair of successive relay images, the visual servoing process can be applied between the successive relay images until the last image. However, such a process is not satisfactory since the camera velocity is null at each transition. To improve the behavior of such a visual-based control scheme, we address the problem of finding realistic image trajectories (i.e corresponding to physically valid camera motion) and corresponding to a minimal camera path between  $N$  given relay images. The obtained image trajectories can then be efficiently tracked using a purely image-based control scheme.

The paper is organized as follow. In Section 2, we recall some fundamentals. In Section 3, a closed-form collineation path between  $N$  relay collineations matrices is obtained. In section 4, the case  $N = 1$ , corresponding to the classical framework in visual servoing, is studied. In Section 5, we describe our approach when relay images are used.

## 2 Fundamentals

### 2.1 The collineation matrix

Consider two views of a scene observed by a camera. A 3-D point  $\mathcal{X}$  with homogeneous coordinates  $\mathbf{X} = [X \ Y \ Z \ 1]^T$  is projected under perspective projection to a point  $\mathbf{x}$  in the first image (with homogeneous coordinates measured in pixel  $\mathbf{x} = [x \ y \ 1]^T$ ) and to a point  $\mathbf{x}^f$  in the second image (with homogeneous coordinates measured in pixel  $\mathbf{x}^f = [x^f \ y^f \ 1]^T$ ). It is well known that there exists a projective homography matrix  $\mathbf{G}$  related to a virtual plane  $\Pi$ , such that for all points  $\mathcal{X}$  (belonging or not to  $\Pi$ )<sup>1</sup>:

$$\mathbf{x} \propto \mathbf{G}\mathbf{x}^f + \tau\mathbf{g}(t) \quad (1)$$

where the matrix  $\mathbf{G}$  is called the collineation matrix,  $\tau$  is a constant scaling factor null if the target point belongs to  $\Pi$  and  $\mathbf{g}(t)$  represents the epipole in the current image that is the projection in the image at time  $t$  of the optical center when the camera is in its desired position. More precisely:

$$\mathbf{g}(t) = \mathbf{K}\mathbf{b}(t) \quad (2)$$

where  $\mathbf{b}(t)$  is the translation vector between the current and desired camera frame (denoted  $\mathcal{F}$  and  $\mathcal{F}^f$  respectively) and

<sup>1</sup> $\mathbf{x} \propto \mathbf{G}\mathbf{x}^f \iff \alpha\mathbf{x} = \mathbf{G}\mathbf{x}^f$  where  $\alpha$  is a scaling factor

$\mathbf{K}$  is the classical non singular matrix containing the intrinsic parameters of the camera. From the knowledge of several matched points, lines or contours [6, 2], it is possible to estimate the collineation matrix and the epipole. For example, if at least four matched points belonging to  $\Pi$  are known,  $\mathbf{G}$  can be estimated by solving a linear system. Else, at least eight points (3 points to define  $\Pi$  and 5 outside of  $\Pi$ ) are necessary to estimate the collineation matrix by using for example the linearized algorithm proposed in [14]. Assuming that the camera calibration is known, the Euclidean homography can be computed up to a scalar factor:<sup>2</sup>

$$\mathbf{H} \propto \mathbf{K}^+ \mathbf{G} \mathbf{K} \quad (3)$$

The Euclidean homography can be decomposed into a rotation matrix and a rank 1 matrix [7]:

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{b}}{d^f} \mathbf{n}^{fT} \quad (4)$$

where  $\mathbf{R}$  represents the rotation matrix between  $\mathcal{F}$  and  $\mathcal{F}^f$ ,  $\mathbf{n}^f$  is the unitary normal to the virtual plane expressed in  $\mathcal{F}^f$  and  $d^f$  is the distance from  $\Pi$  to the origin of  $\mathcal{F}^f$ . From  $\mathbf{G}$  and  $\mathbf{K}$ , it is thus possible to determine the camera motion parameters (i.e the rotation  $\mathbf{R}$  and the scaled translation  $\mathbf{b}_{d^f} = \frac{\mathbf{b}}{d^f}$ ) and the normal vector  $\mathbf{n}^f$  [7]. If the camera is not perfectly calibrated (i.e  $\hat{\mathbf{K}}$  is used instead of  $\mathbf{K}$ ), then, the parameters which can be estimated are [14]:

$$\hat{\mathbf{R}} = \delta\mathbf{K}\mathbf{R}\delta\mathbf{K}^+ \quad (5)$$

$$\hat{\mathbf{n}}^{fT} = \frac{\mathbf{n}^{fT}\delta\mathbf{K}^+}{\|\mathbf{n}^{fT}\delta\mathbf{K}^+\|} \quad (6)$$

$$\hat{\mathbf{b}}_{d^f} = \|\mathbf{n}^{fT}\delta\mathbf{K}^+\|\delta\mathbf{K}\mathbf{b}_{d^f} \quad (7)$$

where  $\delta\mathbf{K} = \hat{\mathbf{K}} - \mathbf{K}$ .

### 2.2 The rotation group $SO(3)$

The group  $SO(3)$  is the set of all  $3 \times 3$  real orthogonal matrices with unit determinant and it has the structure of a Lie group. On a Lie group, the space tangent to the identity has the structure of a Lie algebra. The Lie algebra of  $SO(3)$  is denoted by  $so(3)$ . It consists of the  $3 \times 3$  skew-symmetric matrices, so that the elements of  $so(3)$  are matrices of the form:

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}$$

One of the main connections between a Lie group and its Lie algebra is the exponential mapping. For every  $\mathbf{R} \in SO(3)$ , there exists at least one  $[\boldsymbol{\theta}] \in so(3)$  such that  $e^{[\boldsymbol{\theta}]} = \mathbf{R}$  with (Rodriguez formula):

$$\mathbf{R} = e^{[\boldsymbol{\theta}]} = \mathbf{I} + \frac{\sin\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|}[\boldsymbol{\theta}] + \frac{1 - \cos\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|^2}[\boldsymbol{\theta}]^2 \quad (8)$$

<sup>2</sup> $\mathbf{K}^+$  denotes the inverse of  $\mathbf{K}$

where  $\|\theta\|$  is the standard Euclidean norm. Conversely, if  $\mathbf{R} \in SO(3)$  such that  $\text{Trace}(\mathbf{R}) \neq -1$  then:

$$[\theta] = \log(\mathbf{R}) = \frac{\theta}{2 \sin \theta} (\mathbf{R} - \mathbf{R}^T) \quad (9)$$

where  $\theta$  satisfies:

$$\theta = \|\theta\| = \arccos \left( \frac{1}{2} (\text{Trace}(\mathbf{R}) - 1) \right) \quad (10)$$

If  $\text{Trace}(\mathbf{R}) = -1$ ,  $\log(\mathbf{R})$  can be obtained noticing that  $\theta = \pm \pi \mathbf{u}$  where  $\mathbf{u}$  is a unit length eigenvector of  $\mathbf{R}$  associated with the eigenvalue 1.

Another important connection between  $so(3)$  and  $SO(3)$  involves angular velocities. If  $\mathbf{R}(t)$  is a curve in  $SO(3)$ , then  $\dot{\mathbf{R}}\mathbf{R}^T$  and  $\mathbf{R}^T\dot{\mathbf{R}}$  are skew-symmetric, and hence elements of  $so(3)$ . The element  $\omega$  of  $so(3)$  such that:

$$[\omega] = \mathbf{R}^T \dot{\mathbf{R}} \quad (11)$$

corresponds to the angular velocity of the rigid body.

### 3 Collineation matrix path

Assume that a set of  $N + 1$  relay images  $\mathcal{I} = \{I_0 \cdots I_N\}$  have been acquired and that some image features can be extracted and matched between two successive images (refer to Fig. 1). Assume also that from the extracted image features, the collineation matrices  $\mathbf{G}_{i,i+1}$  between images  $I_i$  and  $I_{i+1}$  can be computed. The collineation matrix  $\mathbf{G}_{i,N} \propto \mathbf{K}(\mathbf{R}_i + \mathbf{b}_{df_i})\mathbf{K}^+$  (refer to (3) and (4)) between images  $I_i$  and  $I_N$  can easily be obtained noticing that :

$$\mathbf{G}_{i,N} = \mathbf{G}_i = \mathbf{G}_{i,i+1} \mathbf{G}_{i+1,i+2} \cdots \mathbf{G}_{N-1,N} \quad (12)$$

Given a set of  $N + 1$  collineation matrices  $\mathcal{G} = \{\mathbf{G}_{0,N} \cdots \mathbf{G}_{N-1,N}, \mathbf{G}_{N,N}\}$  associated to a set of  $N + 1$  time parameters  $\{t_0 \cdots t_{N-1}, t_N\}$ , we want a continuous and piecewise differentiable matrix function  $\mathbf{G}(t)$  such that  $\mathbf{G}(t_i) = \mathbf{G}_i$  for  $i \in \{0 \cdots N\}$  corresponding to a minimal length camera trajectory. This problem can be formulated as follow (problem **PM**):

$$\text{Find } \mathbf{G}(t) \text{ minimizing } J_i = \int_{t_i}^{t_{i+1}} \mathbf{U}^T \mathbf{U} dt$$

for  $i = 0 \cdots N$  and with  $\mathbf{U} = [\omega^T \mathbf{v}^T]^T$  where  $\omega$  is defined by (11),  $\mathbf{v} = \dot{\mathbf{b}}$ , and with boundary conditions:  $\mathbf{G}(t_i) \propto \mathbf{G}_i$ ,  $\mathbf{G}(t_{i+1}) \propto \mathbf{G}_{i+1}$ .

The solution of the problem **PM** can be obtained using classical optimal control formalism and it is given by [15]:

$$\mathbf{G}(\tau) \propto (1 - \tau)\Phi_{i-1} + \tau\Phi_i + (\mathbf{G}_{i-1} - \Phi_{i-1})\Gamma \quad (13)$$

where  $\tau = \frac{t-t_{i-1}}{t_i-t_{i-1}}$  and:

$$\begin{cases} \Gamma(\theta_i, \tau) = \mathbf{K}e^{([\theta_i]_{\times} \tau)} \mathbf{K}^+ \\ \Phi_i = \mathbf{K} \mathbf{b}_{df_i} \mathbf{n}^{f^T} \mathbf{K}^+ \end{cases} \quad (14)$$

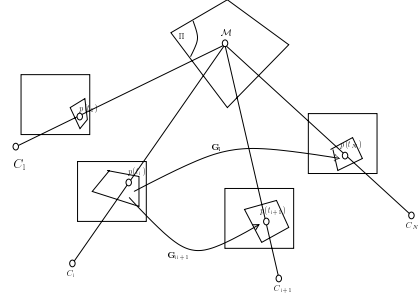


Figure 1: Interpolation of N images

with  $[\theta_i]_{\times} = \log(\mathbf{R}_{i-1}^T \mathbf{R}_i)$ . By introducing the equations (5), (6) and (7) in (13), it can be shown that the path given by (13) is not affected by error on camera intrinsic parameters (the proof can be found in [15]).

### 4 Example: N=1

We assume now that the initial image ( $I_0$  at time  $t = 0$ ) and the desired image ( $I_1$  at time  $t=1$ ) corresponding to the initial and desired robot positions are available. We assume also that some image features can be extracted and matched. This framework is the classical one in visual servoing. From the extracted image features the collineation matrix, at time  $t=0$ ,  $\mathbf{G}_0$ , can be computed. When the desired configuration is reached (at time  $t=1$ ) the collineation matrix is proportional to the identity matrix:  $\mathbf{G}_1 \propto \mathbf{I}$  (corresponding to  $\mathbf{R}_1 = \mathbf{I}$  and  $\mathbf{b}_{df_1} = \mathbf{0}$ ). In this case the path of the collineation matrix is given by (refer to (13) and (14)):

$$\mathbf{G}(t) \propto (1 - t)\Phi_0 + (\mathbf{G}_0 - \Phi_0)\Gamma \quad (15)$$

where :  $\Gamma(\theta_0, t) = \mathbf{K}e^{([\theta_0]_{\times} t)} \mathbf{K}^+$  and  $\Phi_0 = \mathbf{K} \mathbf{b}_{df_0} \mathbf{n}^{f^T} \mathbf{K}^+$  with  $[\theta_0]_{\times} = \log(\mathbf{R}_0^T)$ . The parameters  $\mathbf{n}$ ,  $\mathbf{b}_{df_0}$  and  $\mathbf{R}_0^T$  are obtained from  $\mathbf{G}_0$  using [7]. Note, once again, that the obtained path is not affected by camera calibration errors [18]. The image trajectories can be obtained and followed using an image based control as described in the next sections.

#### 4.1 Trajectories in the image space

In order to control efficiently a robot using visual data, we have to determine the trajectories of some image features in the image space. More precisely, we want to perform smooth trajectories  $\mathbf{s}^*(t) = [x_1^*(t) y_1^*(t) \cdots x_n^*(t) y_n^*(t)]^T$  of  $n$  projected points in the image between a given start point  $\mathbf{s}^*(0) = [x_1^*(0) y_1^*(0) \cdots x_n^*(0) y_n^*(0)]^T$  and a given desired point  $\mathbf{s}^*(1) = [x_1^*(1) y_1^*(1) \cdots x_n^*(1) y_n^*(1)]^T$  (refer to Figure 2). We denote  $\mathbf{x}_i^*(t) = [x_i^*(t) y_i^*(t) 1]^T$  the vector of homogeneous coordinates expressed in pixel of the projection of a 3-D point  $\mathcal{X}_i$  in the current desired image (at time  $t$ ).

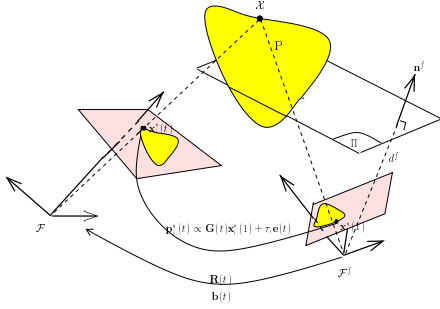


Figure 2: Features trajectories in the image

We define vector  $\mathbf{h}_i = \alpha_i(t)\mathbf{x}_i^*(t)$  such that:

$$\mathbf{h}_i(t) = \alpha_i(t)\mathbf{x}_i^*(t) = \mathbf{G}(t)\mathbf{x}_i^*(1) + \tau_i\mathbf{g}(t) \quad (16)$$

where  $\alpha_i(t)$  is a positive scaling factor depending on time,  $\tau_i$  is a constant scaling factor null if the target point belongs to  $\Pi$ . After that the initial collineation has been estimated, the optimal path of the collineation matrix can be computed as described previously. The initial value of the epipole,  $\mathbf{g}(0) = \mathbf{g}_0$ , can also be computed directly from image data (i.e,  $\mathbf{g}_0$  is independent of the  $\mathbf{K}$ -matrix). Furthermore, it is easy to show that the optimal trajectories of the epipole, with respect to the previously cited criteria, are of the form [15]:

$$\mathbf{g}(t) = (1-t)\mathbf{g}_0 \quad (17)$$

Such trajectories of the epipole are not affected by error on intrinsic parameters. Note also that the scaling factor  $\tau_i$  is not time dependent and can be computed directly from the initial and desired image data since (refer to (16)):

$$\alpha_i(t)\mathbf{x}_i^*(0) \wedge \mathbf{x}_i^*(0) = 0 = \mathbf{G}(t)\mathbf{x}_i^*(1) \wedge \mathbf{x}_i^*(0) + \tau_i\mathbf{g}(t) \wedge \mathbf{x}_i^*(0)$$

We thus obtain <sup>3</sup>:  $\tau_i = -\frac{(\mathbf{G}_0\mathbf{x}_i^*(1) \wedge \mathbf{x}_i^*(0))_1}{(\mathbf{g}_0 \wedge \mathbf{x}_i^*(0))_1}$

The vector  $\mathbf{h}_i$  is not affected by error on intrinsic parameters since  $\mathbf{G}(t)$ ,  $\mathbf{e}(t)$  and  $\tau_i$  ( $\forall i \in \{1 \dots n\}$ ) can be computed without error even if the  $\mathbf{K}$ -matrix is unknown. The trajectories of the considered point in the image corresponding to an optimal camera path can thus also be computed without error, using:

$$x_i^*(t) = \frac{(\mathbf{h}_i(t))_1}{(\mathbf{h}_i(t))_3} \quad y_i^*(t) = \frac{(\mathbf{h}_i(t))_2}{(\mathbf{h}_i(t))_3} \quad (18)$$

## 4.2 Control scheme

To track the trajectories using an image-based control scheme, we use the task function approach introduced by

<sup>3</sup>( $\mathbf{v}$ )<sub>j</sub> denotes the  $j^{th}$  components of  $\mathbf{v}$

Samson et al in [20]. A vision-based task function  $\mathbf{e}$  to be regulated to  $\mathbf{0}$  is defined by [5]:

$$\mathbf{e} = \hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}^*(t)) \quad (19)$$

The time varying vector  $\mathbf{s}^*(t)$  is the desired trajectory of  $\mathbf{s}$  computed as previously explained (more generally, we use the notation  $x^*(t)$  to represent the planned parameter  $x$ ). The matrix  $\mathbf{L}$  denotes the interaction matrix related to  $\mathbf{s}$  (also called image Jacobian). It links the variation of the visual features with respect to the camera velocity  $\mathbf{T}_c$  with  $\dot{\mathbf{s}} = \mathbf{L}\mathbf{T}_c$ . The matrix  $\hat{\mathbf{L}}^+$  is the pseudo-inverse of a chosen model of  $\mathbf{L}$ . An exponential decay of  $\mathbf{e}$  toward  $\mathbf{0}$  can be obtained by imposing  $\dot{\mathbf{e}} = -\lambda\mathbf{e}$  ( $\lambda$  being a proportional gain), the corresponding control law is:

$$\mathbf{T}_c = -\lambda\mathbf{e} - \frac{\partial\mathbf{e}}{\partial t} \quad (20)$$

Using such control law, a well known sufficient condition to ensure global asymptotic stability of the system is [20]:

$$\hat{\mathbf{L}}^+\mathbf{L} > 0 \quad (21)$$

For a point  $\mathcal{X}$  with coordinates  $[X \ Y \ Z]^T$  in the current camera frame and image coordinates  $\mathbf{x} = [x \ y \ 1]^T$  in pixels, the interaction matrix  $\mathbf{L}(\mathbf{x}, Z)$  related to  $\mathbf{x}$  is given by:

$$\mathbf{a} \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{m_x}{Z} & m_x m_y & -(1+m_x^2) & m_y \\ 0 & -\frac{1}{Z} & \frac{m_y}{Z} & (1+m_y^2) & -m_x m_y & -m_x \end{bmatrix}$$

with  $\mathbf{m} = [m_x \ m_y \ 1]^T = \mathbf{K}^+\mathbf{x}$  and:

$$\mathbf{a} = \begin{bmatrix} fp_u & -fp_u \cot(\theta) \\ 0 & fp_v / \sin(\theta) \end{bmatrix}$$

where  $f$  is the camera focal length,  $p_u$  and  $p_v$  are the magnifications respectively in the  $u$  and  $v$  directions, and  $\theta$  is the angle between these axes. When  $\mathbf{s}$  is composed of the image coordinates  $\mathbf{x}_i$  of  $n$  points, the corresponding interaction matrix is:

$$\mathbf{L}(\mathbf{s}, \mathbf{Z}) = [\mathbf{L}^T(\mathbf{x}_1, Z_1) \dots \mathbf{L}^T(\mathbf{x}_n, Z_n)]^T \quad (22)$$

A classical choice for  $\hat{\mathbf{L}}$  is  $\mathbf{L}(\mathbf{s}(1), \hat{\mathbf{Z}}(1))$  (that is the value of  $\mathbf{L}$  at the final desired position), in this case, condition (21) is ensured only in a neighborhood of the desired position [1]. We will use the value of  $\mathbf{L}$  at the current desired position for  $\hat{\mathbf{L}}$  (that is  $\hat{\mathbf{L}} = \mathbf{L}(\mathbf{s}^*(t), \hat{\mathbf{Z}}^*(t))$  rather than  $\mathbf{L}(\mathbf{s}(1), \hat{\mathbf{Z}}(1))$ . With this choice, condition (21) is ensured along the planned trajectories and not only in a neighborhood of the final desired position.

The interaction matrix depends of the  $\hat{\mathbf{Z}}^*$ -vector. This vector can be rewritten as  $\hat{\mathbf{Z}}^*(t) = \hat{d}^f \Sigma^*(t)$  where  $\hat{d}^f$  is an approximated value of  $d^f$  (that is the only parameter that has

to be introduced “by hand”) and  $\Sigma^*(t) = [\rho_1^*(t) \cdots \rho_n^*(t)]$ , where [15]:

$$\begin{cases} \hat{\rho}_i^*(t) = \frac{\det(\mathbf{G}^*(t)) \det(\mathbf{G}_0 - \Phi_0)^{-2/3}}{(\hat{\mathbf{K}}^+ (\mathbf{G}_0 - \Phi_0) \Gamma \hat{\mathbf{K}} \hat{\mathbf{n}}^f)^T \hat{\mathbf{K}}^+ \mathbf{x}_i^*(t)} & \text{if } \mathcal{X}_i \in \Pi \\ \hat{\rho}_i^*(t) = \frac{(1-t)\hat{\beta}(t) \|\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f\|}{\|\hat{\beta}(t) \hat{\mathbf{K}}^+ \mathbf{p}_i^f - \hat{\mathbf{K}}^+ (\mathbf{G}_0 - \Phi_0) \Gamma^*(t) \mathbf{p}_i^*(t)\|} & \text{if } \mathcal{X}_i \notin \Pi \end{cases}$$

in which:

$$\hat{\beta}(t) = \frac{\|[\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f]_{\times} (\mathbf{G}_0 - \Phi_0) \Gamma^*(t) \mathbf{x}_i^*(1)\|}{\|[\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f]_{\times} \hat{\mathbf{K}}^+ \mathbf{x}_i^*(t)\|}$$

If the target is known to be motionless, we have  $\frac{\partial \mathbf{e}}{\partial t} = -\hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t}$  and the control law (20) can be rewritten as follows:

$$\mathbf{T}_c = -\lambda \mathbf{e} + \hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t} \quad (23)$$

where the term  $\hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t} = \hat{\mathbf{L}}^+ [\frac{\partial x_1^*}{\partial t} \frac{\partial y_1^*}{\partial t} \cdots \frac{\partial x_n^*}{\partial t} \frac{\partial y_n^*}{\partial t}]^T$  allows to compensate the tracking error. More precisely, we have from (16):

$$\frac{\partial \mathbf{x}_i^*}{\partial t} = \frac{1}{\alpha_i(t)} \left[ \frac{\partial \mathbf{G}}{\partial t} \mathbf{x}_i^*(1) + \beta_i \frac{\partial \mathbf{g}}{\partial t} - \frac{\partial \alpha_i}{\partial t} \mathbf{x}_i^*(t) \right] \quad (24)$$

and if we rewrite the collineation and the epipole as follows:

$$\mathbf{G}(t) = \begin{bmatrix} \mathbf{G}_1(t) \\ \mathbf{G}_2(t) \\ \mathbf{G}_3(t) \end{bmatrix} \quad \mathbf{g}(t) = \begin{bmatrix} \mathbf{g}_1(t) \\ \mathbf{g}_2(t) \\ \mathbf{g}_3(t) \end{bmatrix}$$

we obtain from (15), (16) and (17):

$$\begin{cases} \frac{\partial \mathbf{G}}{\partial t} = -[-\Phi_0 + (\mathbf{G}_0 + \Phi_0) \Gamma(t)] = \begin{bmatrix} \frac{\partial \mathbf{G}_1}{\partial t} \\ \frac{\partial \mathbf{G}_2}{\partial t} \\ \frac{\partial \mathbf{G}_3}{\partial t} \end{bmatrix} \\ \frac{\partial \mathbf{g}}{\partial t} = -\mathbf{g}_0 = \begin{bmatrix} \frac{\partial \mathbf{g}_1}{\partial t} \\ \frac{\partial \mathbf{g}_2}{\partial t} \\ \frac{\partial \mathbf{g}_3}{\partial t} \end{bmatrix} \\ \alpha_i(t) = \mathbf{G}_3(t) \mathbf{x}_i^*(1) + \tau_i \mathbf{g}_3(t) \\ \frac{\partial \alpha_i}{\partial t} = \frac{\partial \mathbf{G}_3}{\partial t} \mathbf{x}_i^*(1) + \tau_i \frac{\partial \mathbf{g}_3}{\partial t} \end{cases}$$

The term  $\frac{\partial \mathbf{s}^*}{\partial t}$  is finally obtained by introducing the previous relations in (24).

### 4.3 Experimental results

The proposed method has been tested in a positioning task with respect to unknown scenes using a CCD camera mounted on a six degree of freedom robot arm. In the first experiment, the target is a non-planar object composed by nine white marks. The extracted visual features are the image coordinates of the center of gravity of each mark. The

desired images have been acquired during an off-line step. The algorithm proposed in [14] has been used to obtain the initial collineation. The images corresponding to the desired and initial camera positions are given in the Figures 3(a) and 3(b) respectively. The corresponding camera displacement is very important ( $b_x = -195mm, b_y = -610mm, b_z = -1455mm, (\mathbf{u}\theta)_x = -68dg, (\mathbf{u}\theta)_y = -41dg, (\mathbf{u}\theta)_z = -144dg$ ). In order to check the robustness with respect to modeling errors of the proposed approach, two different sets of parameters have been used, **correct calibration**: the correct intrinsic parameters and the correct value of  $d^f$  (that is 36 cm) have been used (see Figure 4); **bad calibration**: an error of 50% has been added on the intrinsic parameters, while  $d^f$  has been set to 80 cm (see Figure 5).

**1) Correct calibration:** Planned and tracked trajectories are plotted in Figures 4(a) and 4(b) respectively. We first note that the tracked trajectories and the planned trajectories are almost similar. This shows the efficiency of the proposed control scheme. The tracking error ( $\mathbf{s}(t) - \mathbf{s}^*(t)$ ) is plotted in Figure 4(d), and it confirms the previous comment since the maximal error remains small (always less than 5 pixels). The error on the coordinates of each target point between its current and its desired location in the image ( $\mathbf{s}(t) - \mathbf{s}^*(1)$ ) is given in Figure 4(c). The convergence of the coordinates to their desired value demonstrates the correct realization of the task. The computed control law is given in Figure 4(e). Note finally that the camera optical center move along a straight line as can be seen in Figure 4(f).

**1) Bad calibration:** First, we note that the planned and followed trajectories obtained with or without modeling errors are almost similar (refer to Figures 4 and 5). That confirms the robustness of the path planning and of the control scheme with respect to calibration errors and errors on  $d^f$ . Once again, as can be seen in Figures 5(a) and 5(b) the planned and the tracked trajectories are also similar. The tracking error, given in Figure 5(e), remains small during the serving (less than 8 pixels). We note also the stability and the robustness of the control law (see Figures 5(e)). Once again, the task is correctly realized. This is shown by the convergence of the image points coordinates to their desired value (refer to Figure 5(d)).

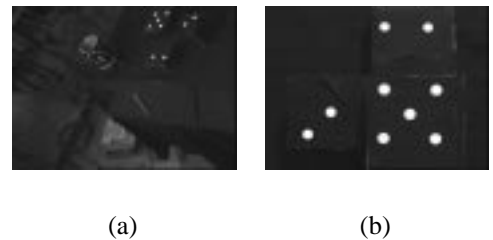


Figure 3: (a) Initial image and (b) desired image

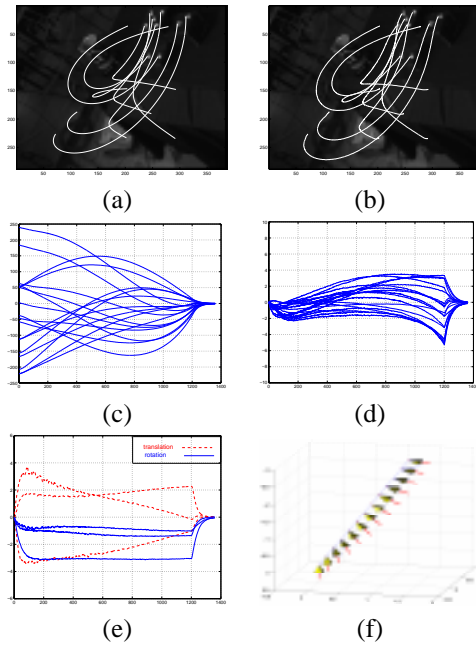


Figure 4: Correct calibration: (a) planned trajectories, (b) followed trajectories, (c) error in image points coordinates (pixels), (d) tracking error (pixels) (e) velocities (cm/s and dg/s) and (f) camera trajectory.

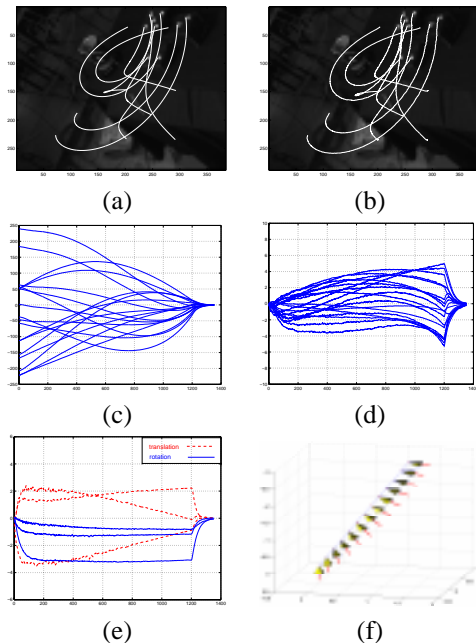


Figure 5: Bad calibration: (a) planned trajectories, (b) followed trajectories, (c) error in image points coordinates (pixels), (d) tracking error (pixels) (e) velocities (cm/s and dg/s) and (f) camera trajectory.

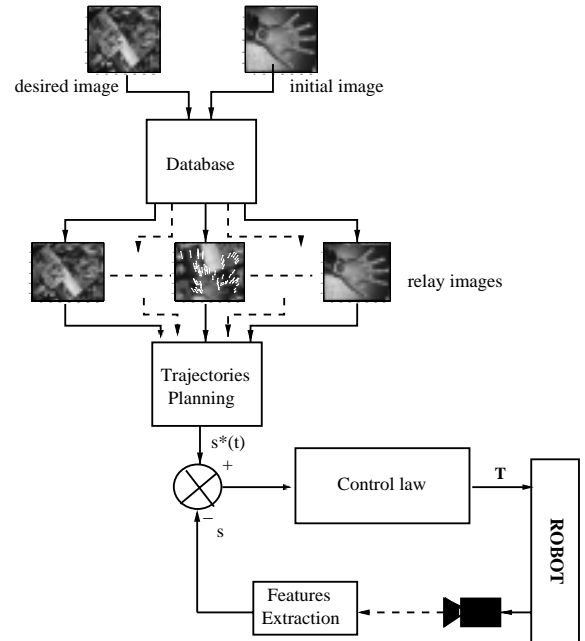


Figure 6: Block diagram of the interpolation and tracking process

## 5 Case of N images

Classical visual servoing techniques make assumptions on the link between the initial and target images, limiting the applicability of these techniques to relatively small displacement when the scene is complex. Indeed, if a sufficient number of image features can not be matched in these images, the visual servoing can not be realized. We propose to use relay images to cope with this limitation.

### 5.1 Obtaining the relay images

Recent work in image database analysis have emerged as solution to the problem of retrieving and delivering images from large database using query [4]. In our lab, these techniques have been investigated [8]. We use these methods to obtain the relay images between the initial and target images. In a first off-line step, the robot acquires a large set of images of its workspace. Ideally, these images must provide a representative sample of all the points of view which could be reached during the operational phase. Points of interest of all these images are extracted (using the Harris detector [9]) and some invariants are computed [21]. That allows on the one hand to index these images in a database which will be used to retrieve fastly images acquired during the operational phase, and on the other hand to match images by pair. A graph is then derived from this matching. The nodes of the graph are the images. An edge between two images indicates that the images could be matched. The edges are valued in a way inversely proportional to the number of matched image features (the matching is realized using the

algorithm proposed in [24]).

In the second on-line step, the robot acquires an initial image at an unspecified place of its workspace. A task is specified in the form of an image to reach. The system then seeks in the image database the closest images to the initial and desired images. The shortest path between these images in the graph is then obtained by using the Dijkstra's algorithm. We thus obtain an ordered set of relay images such that between two successive images of this set a sufficient number of image features can be matched. As already stated, classical visual servoing could be used between two successive images until the last image. However, using such a process, the camera velocity would be null at each transition. That is why the trajectories in the images are planned. The features are interpolated as for the case  $N=1$  (see Section 4.1) and the displacement is then carried out using the control scheme described in Section 4.2 (see Fig. 6).

## 5.2 Experimental results

In this section, our approach is validated by realizing a positioning task. The images corresponding to the desired and initial camera positions are given in the Figs. 7(a) and 7(b). In this case, the SFMJ is impossible to realize. However, from the graph built with the image database, eight relay images are obtained (see 8). The trajectories of interesting points are then planned. The planned and followed trajectories are given in the Figs. 8 and 9. We note that these trajectories are similar. The camera trajectory is given in the Fig. 10(a). The tracking error (defined as  $\frac{1}{2n} \sum_1^n (x_i(t) - x_i^*(t)) + (y_i(t) - y_i^*(t))$  and plotted in Fig. 10(b)) remains sufficiently small (always less than 5 pixels) to ensure a good behavior of the control scheme.

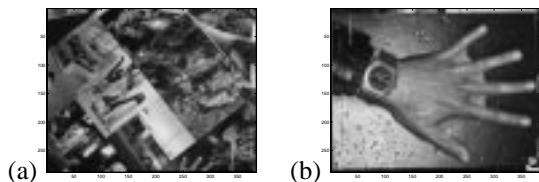


Figure 7: (a) Initial image and (b) desired image

## 6 Conclusion

In this paper, we have addressed the problem of finding and tracking image trajectories of visual features between  $N$  relay images automatically extracted from a database. The obtained camera trajectory corresponds to a minimal path. The method is model-free and it does not need any accurate calibration. By coupling the path planning step with an image-based servoing, the proposed method improves significantly the behavior of image-based servoing. We have validated our approach in a robotic platform by realizing positioning tasks with respect to an unknown scene. Future work will be devoted to introduce nonholonomic constraints in the planned trajectories.



Figure 8: Planned trajectories



Figure 9: Followed trajectories



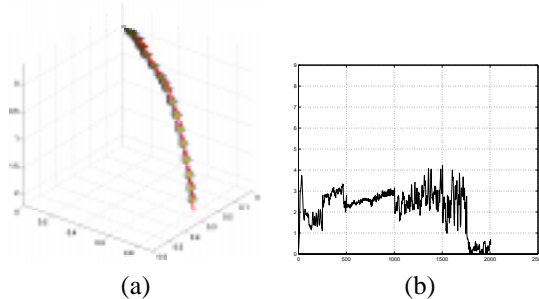


Figure 10: (a) Camera trajectory (m), (b) Tracking error (pixels)

## References

- [1] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. *The Confluence of Vision and Control*, D. Kriegman, G. Hager, A. Morse (eds), LNCIS Series, Springer Verlag, 237:66–78, 1998.
- [2] G. Chesi, E. Malis, and R. Cipolla. Automatic segmentation and matching of planar contours for visual servoing. In *International Conference on Robotics and Automation*, San Francisco, April 2000.
- [3] N.J. Cowan and D.E. Koditschek. Planar image based visual servoing as a navigation problem. In *IEEE Int. Conf. on Robotics and Automation*, pages 611–617, Detroit, Michigan, May 1999.
- [4] M. De Marsico, L. Cinque, and S. Levialdi. Indexing pictorial documents by their content : a survey of current techniques. *IVC*, 15(2):119–141, 1997.
- [5] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.
- [6] O. Faugeras. *Three-dimensionnal computer vision : a geometric view point*. MIT press, Cambridge, Massachusetts, 1993.
- [7] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.
- [8] P. Gros, Anthony Remazeilles, and F. Chaumette. Les invariants: introduction et application à la vision et à la robotique. In *Journées Nationales de la Recherche en Robotique*, Presqu'île de Giens, France, October 2001.
- [9] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Conference*, pages 189–192, 1988.
- [10] K. Hashimoto. *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapor, 1993.
- [11] K. Hashimoto and T. Noritugu. Potential switching control in visual servo. In *IEEE Int. Conf. on Robotics and Automation*, pages 2765–2770, San Francisco, USA, April 2000.
- [12] K. Hosoda, K. Sakamoto, and M. Asada. Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1(3):29–34, August 1995.
- [13] 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, October 1996.
- [14] E. Malis and F. Chaumette. 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97, June 2000.
- [15] Y. Mezouar. *Planification de trajectoires pour l'asservissement visuel*. PhD thesis, Université de Rennes 1, IRISA, 2001.
- [16] Y. Mezouar and F. Chaumette. Design and tracking of desirable trajectories in the image space by integrating mechanical and visibility constraints. In *IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 731–736, Seoul, Korea, May 2001.
- [17] Y. Mezouar and F. Chaumette. Model-free optimal trajectories in the image space. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'01*, Maui, Hawaii, October 2001.
- [18] Y. Mezouar and F. Chaumette. Model-free optimal trajectories in the image space: Application to robot vision control. In *IEEE Int. Conference on Computer Vision and Pattern Recognition, CVPR'01*, Kauai Marriott, Hawaii, December 2001.
- [19] A. Ruf and R. Horaud. Visual trajectories from uncalibrated stereo. *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 83–91, September 1997.
- [20] C. Samson, B. Espiau, and M. Le Borgne. *Robot Control : The Task Function Approach*. Oxford University Press, 1991.
- [21] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, May 1997.
- [22] R. Singh, R. M. Voyle, D. Littau, and N. P. Papanikolopoulos. Alignment of an eye-in-hand system to real objects using virtual images. *Workshop on Robust Vision for Vision-Based Control of Motion, IEEE Int. Conf. on Robotics and Automation*, May 1998.
- [23] Hong Zhang and Jim Ostrowski. Optimal motion planning in the image plane for mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, San Francisco, USA, April 2000.
- [24] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, October 1995.