
Youcef Mezouar*
François Chaumette

IRISA / INRIA Rennes
Campus de Beaulieu
35042 Rennes cedex, France

Optimal Camera Trajectory with Image-Based Control

Abstract

Image-based servo is a local control solution. Thanks to the feedback loop closed in the image space, local convergence and stability in the presence of modeling errors and noise perturbations are ensured when the error is small. The principal deficiency of this approach 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. In this paper we address the problem of finding realistic image-space trajectories that correspond to optimal 3D trajectories. The camera calibration and the model of the observed scene are assumed unknown. First, a smooth closed-form collineation path between given start and end points is obtained. This path is generated in order to correspond to an optimal camera path. The trajectories of the image features are then derived and efficiently tracked using a purely image-based control. A second path planning scheme, based on the potential field method is briefly presented. It allows us to introduce constraints in the desired trajectory to be realized. Such constraints are, for instance, to ensure that the object of interest remains in the camera field of view and to avoid the robot joints limits. Experimental results obtained on a six-degrees-of-freedom eye-in-hand robotic system are presented and confirm the validity of the proposed approach.

KEY WORDS—visual servoing, path planning, path tracking

1. Introduction

Image-based servoing is now a well-known local control framework (Hashimoto 1993; Hutchinson, Hager, and Corke 1996). In this approach, the reference image of the object corresponding to a desired position of the robot is 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 robot (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 singularities in the relation from the image space to the workspace (Chaumette 1998). Dealing with this deficiency, 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 Hashimoto and Noritugu (2000), relay images that interpolate initial and reference image features using an affine approximation of the relationship between initial and desired images, coupled to a potential switching control scheme, are proposed to enlarge the stable region. In Hosoda, Sakamoto, and Asada (1995), a trajectory generator using a stereo system is proposed and applied to obstacle avoidance. An alignment task for a four-degrees-of-freedom (4-DoF) robot using intermediate views of the object synthesized by image morphing is presented in Singh et al. (1998). A path planning for a straight-line robot translation observed by a weakly calibrated stereo system is performed in Ruf and Horaud (1997). In previous work (Mezouar and Chaumette 2002), 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, Lopes, and Koditschek (2000) describe a globally stabilizing method using navigation function for eye-to-hand setup. However,

*Current address: LASMEA-CNRS UMR 6602, Université Blaise Pascal, 63177 Aubière-France Youcef.Mezouar@lasmea.univ-bpclermont.fr
The International Journal of Robotics Research
Vol. 22, No. 10–11, October–November 2003, pp. 781–803,
©2003 Sage Publications

none of these works has dealt with optimality issues. In Zhang and Ostrowski (2000), a numerical framework for the design of optimal trajectories in the image space is described and applied to the simple case of a one-dimensional (1D) camera in a two-dimensional (2D) workspace. In Mezouar and Chaumette (2001), our preliminary results have been presented. This paper gives an analytical solution to optimal path planning in the image space for a general setup. Additionally, the CAD model of the observed target and the camera calibration are not assumed known.

On the other hand, a new control scheme for a general setup, called $2\frac{1}{2}$ D visual servoing, has been proposed in Malis, Chaumette, and Boudet (1999) which globally stabilizes the system. In this approach, the error function to be regulated to zero is composed of the camera orientation parameters and image features. However, $2\frac{1}{2}$ D visual servoing techniques are not purely image-based, thus the robustness of image-based techniques are not fully obtained. In particular, the sensitivity to measurement perturbation due to the computation of a homography matrix at each iteration of the control loop to extract the Cartesian part of the error function can be problematic close to the convergence.

In this paper, we address the problem of finding closed-form image trajectories between the initial and desired images corresponding to an optimal camera path with respect to minimum energy and acceleration criteria. The obtained image trajectories can be efficiently tracked using a purely image-based control scheme. Furthermore, classical visual servoing techniques make assumptions about the link between the initial image and the desired one. When the camera displacement between the acquisitions of the initial and desired images is large and/or when the observed scene is complex, the steps of finding and matching joint image features (SFMJF) can be difficult and even sometimes impossible if no feature belongs to both images. In such case, the servoing can not be realized. A possible solution is to use a set of relay images (such that between two successive SFMJF are feasible) extracted, for example, from an image database obtained and indexed off-line. To cope with this framework, we also show how our strategy can be generalized to the case where $N + 1$ images are available. Additionally, we briefly present a second path planning scheme based on the potential field method. It allows us to introduce constraints in the desired trajectory to be realized. Such constraints are, for instance, to ensure that the object of interest remains in the camera field of view and to avoid the robot joints limits. In counterpart, the analytical forms of the trajectories in the image space are no longer available, and the corresponding camera trajectory deviates from the optimal one.

The paper is organized as follows. In Section 2, we recall some fundamentals. In Section 3, we address the problem of finding a closed-form smooth collineation path between given start and end points and corresponding to optimal camera trajectories. In this section, we first study the case where the

camera is displaced according to a general rigid motion and then two interesting particular cases are addressed (namely pure rotational and pure translational motions). In Section 4, the optimal collineation path is used to determine the optimal path in the image, and an example illustrating our approach is presented. In Section 5, we show how to use an image-based approach to track the trajectories in the image space. Experimental results using a 6-DoF eye-in-hand robotic platform are presented in Section 6. In Section 7, our strategy is generalized to the case where a set of images is available. Finally, in Section 8, we briefly describe a second path planning scheme able to take into account constraints on the image trajectories.

2. Fundamentals

2.1. The Collineation Matrix

Consider two views of a scene observed by a camera. A 3D 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 Π , such that for all points \mathcal{X} belonging to Π , $\mathbf{x} \propto \mathbf{G}\mathbf{x}^f$.¹ When \mathbf{x} and \mathbf{x}^f are expressed in pixels, matrix \mathbf{G} is called the collineation matrix. From the knowledge of several matched points, lines or contours (Faugeras 1993; Chesi, Malis, and Cipolla 2000), it is possible to estimate the collineation matrix. For example, if at least four points belonging to Π are matched, \mathbf{G} can be estimated by solving a linear system. Otherwise, at least eight points (three points to define Π and five outside of Π) are necessary to estimate the collineation matrix by using, for example, the linearized algorithm proposed in Malis and Chaumette (2000). If the camera calibration is known, the Euclidean homography can be computed up to a scalar factor²

$$\mathbf{H} \propto \mathbf{K}^+ \mathbf{G} \mathbf{K} \quad (1)$$

where \mathbf{K} is a non-singular matrix containing the intrinsic parameters of the camera

$$\mathbf{K} = \begin{bmatrix} fp_u & -fp_u \cot(\alpha) & u_0 \\ 0 & fp_v / \sin(\alpha) & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a} & u_0 \\ & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Here, u_0 and v_0 are the pixel coordinates of the principal point, f is the focal length, p_u and p_v are the magnifications respectively in the u and v directions, and α is the angle between these axes. The Euclidean homography can be decomposed

1. $\mathbf{x} \propto \mathbf{G}\mathbf{x}^f \iff \alpha\mathbf{x} = \mathbf{G}\mathbf{x}^f$ where α is a scaling factor.

2. \mathbf{K}^+ denotes the inverse of \mathbf{K} .

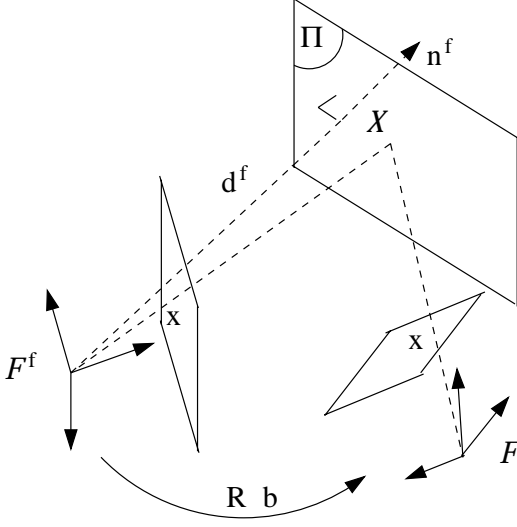


Fig. 1. Geometry of two views.

into a rotation matrix and a rank 1 matrix (Faugeras and Lustman 1988)

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{b}}{d^f} \mathbf{n}^{fT} \quad (2)$$

where \mathbf{R} and \mathbf{b} represent the rotation matrix and the translation vector between the current and the desired camera frames (denoted by \mathcal{F} and \mathcal{F}^f respectively), \mathbf{n}^f is the unitary normal to the virtual plane expressed in \mathcal{F}^f , and d^f is the distance from Π to the origin of \mathcal{F}^f (see Figure 1).

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 $\mathbf{b}_{df} = \frac{\mathbf{b}}{d^f}$) and the normal vector \mathbf{n}^f , by using for example one of the algorithms proposed in Faugeras and Lustman (1988) or Zhang and Hanson (1995). The ratio ρ between the Z -coordinate and d^f , which we will use in the sequel, can also be determined

$$\begin{cases} \rho = \frac{Z}{d^f} = \frac{r}{\mathbf{n}^T \mathbf{m}} & \text{if } \mathcal{X} \in \Pi \\ \rho = \frac{Z}{d^f} = \beta \frac{\|\mathbf{b}_{df}\|}{\|\beta \mathbf{m} - \mathbf{R}\|} & \text{if } \mathcal{X} \notin \Pi \end{cases} \quad (3)$$

where $\mathbf{m} = \mathbf{K}^+ \mathbf{x}$, $\mathbf{n} = \mathbf{R} \mathbf{n}^f$, $r = 1 + \mathbf{n}^{fT} \mathbf{R} \mathbf{b}_{df}$, $\beta = \frac{\|[\mathbf{b}] \times \mathbf{R} \mathbf{m}\|}{\|[\mathbf{b}] \times \mathbf{m}\|}$. If the camera is not perfectly calibrated (i.e., $\hat{\mathbf{K}}$ is used instead of \mathbf{K}), then the estimated homography matrix is

$$\hat{\mathbf{H}} = \hat{\mathbf{K}} \mathbf{G} \hat{\mathbf{K}}^+ = \delta \mathbf{K}^+ \mathbf{G} \delta \mathbf{K}$$

where $\delta \mathbf{K} = \hat{\mathbf{K}}^+ \mathbf{K}$. Matrix $\hat{\mathbf{H}}$ can be decomposed as the sum of a matrix similar to a rotation matrix and of a rank 1 matrix

$$\hat{\mathbf{H}} = \hat{\mathbf{H}}_\infty + \hat{\mathbf{b}}_{df} \hat{\mathbf{n}}^{fT}$$

with

$$\hat{\mathbf{H}}_\infty = \hat{\mathbf{R}} = \delta \mathbf{K} \mathbf{R} \delta \mathbf{K}^+. \quad (4)$$

The matrix $\hat{\mathbf{H}}_\infty = \hat{\mathbf{R}}$ is called homography at infinity since it maps projected points lying in the plane at infinity. Furthermore, if the collineation at infinity \mathbf{G}_∞ is available, we have

$$\hat{\mathbf{n}}^{fT} = \frac{\mathbf{n}^{fT} \delta \mathbf{K}^+}{\|\mathbf{n}^{fT} \delta \mathbf{K}^+\|} \quad (5)$$

$$\hat{\mathbf{b}}_{df} = \|\mathbf{n}^{fT} \delta \mathbf{K}^+\| \delta \mathbf{K} \mathbf{b}_{df}. \quad (6)$$

This matrix \mathbf{G}_∞ can be determined using, for example, the methods described in Robert et al. (1995) or Viéville, Zeller, and Robert (1996). These methods can be used even if the camera parameters are unknown if some knowledge about the scene is available (for example, if the camera observes three points at infinity or a reference plane parallel to the image plane). In practice, the algorithm proposed in Malis and Chaumette (2000) has been used to obtain the initial collineation matrix, from which the collineation matrix at infinity between the initial and desired images is extracted using the algorithm proposed in Faugeras and Lustman (1988). This algorithm is based on the fundamental relation (2). It was thus designed for the case where the camera calibration is known. This means that the relations (5) and (6) are not exactly verified when the calibration is unknown. However, we will see in the experiments that, even if this algorithm is used in an uncalibrated framework, the results are very satisfactory. Let us finally note that, when the camera motion is a pure translation, the homography matrix at infinity is the identity matrix. Furthermore, when the camera motion is a pure rotation, $\hat{\mathbf{H}}_\infty$ corresponds to the homography matrix $\hat{\mathbf{H}}$.

2.2. Brief Review of $SO(3)$

The group $SO(3)$ is the set of all 3×3 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 3×3 skew-symmetric matrices, so that the elements of $so(3)$ are matrices of the form

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}.$$

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)

$$\mathbf{R} = e^{[\boldsymbol{\theta}]} = \mathbf{I} + \frac{\sin \|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|} [\boldsymbol{\theta}] + \frac{1 - \cos \|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|^2} [\boldsymbol{\theta}]^2 \quad (7)$$

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

$$[\boldsymbol{\theta}] = \log(\mathbf{R}) = \frac{\theta}{2 \sin \theta} (\mathbf{R} - \mathbf{R}^T) \quad (8)$$

where θ satisfies

$$\theta = \|\boldsymbol{\theta}\| = \arccos\left(\frac{1}{2}(\text{Trace}(\mathbf{R}) - 1)\right). \quad (9)$$

If $\text{Trace}(\mathbf{R}) = -1$, $\log(\mathbf{R})$ can be obtained noticing that $\boldsymbol{\theta} = \pm\pi \mathbf{u}$ 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 the element of $so(3)$. The element $\boldsymbol{\omega}$ of $so(3)$ such that

$$[\boldsymbol{\omega}] = \mathbf{R}^T\dot{\mathbf{R}} \quad (10)$$

corresponds to the angular velocity of the rigid body.

In the following, we consider that some image features (points, lines, contours) can be extracted and matched from the initial image and a desired image of the scene. 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 (Faugeras 1993; Chesi, Malis, and Cipolla 2000). Note also that, when the desired configuration is reached (at time $t = 1$), the collineation matrix is proportional to the identity matrix: $\mathbf{G}^f \propto \mathbf{I}$. In the following sections, we consider the problem of finding a smooth path of the collineation matrix between \mathbf{G}_0 and \mathbf{G}^f corresponding to an optimal camera path with respect to the criterion which will be specified in the following. The image trajectories are then derived from the collineation path and tracked using an image-based strategy.

3. Optimal Collineation Trajectories

The case where the camera is displaced according to a general rigid motion is first studied and then two particular cases are addressed (namely, pure rotational and pure translational motions). In the first case, we will see that the obtained trajectories in the image space are independent of the camera internal parameters if the collineation matrix at infinity between the initial and final images is known. If the motion is a pure rotation or a pure translation, then this assumption is not necessary to ensure the independence with respect to camera calibration.

3.1. General Camera Motions

The current position of the camera with respect to its desired position is defined by the rotation matrix $\mathbf{R}(t)$ and the translation vector $\mathbf{b}(t)$. The collineation matrix is then given by

$\mathbf{G}(t) \propto \mathbf{K}^+ (\mathbf{R} + \mathbf{b}_d \mathbf{n}^f)^T \mathbf{K}$. We denote by \mathbf{U} the 6×1 vector $[\mathbf{v}^T \boldsymbol{\omega}^T]^T$, where \mathbf{v} denotes the time derivative of \mathbf{b} and $\boldsymbol{\omega}$ is defined by eq. (10). We consider the problems, denoted by **PC1** and **PC2**, of finding a path of the collineation matrix corresponding to the minimum energy and minimum acceleration problem, respectively. These problems can be formulated as follows.

(PC1) Find $\mathbf{G}(t)$ minimizing

$$J_1 = \int_0^1 \mathbf{U}^T \mathbf{U} dt$$

subject to eq. (10), $\mathbf{v} = \dot{\mathbf{b}}$ and with boundary conditions:

$$\mathbf{G}(0) \propto \mathbf{G}_0$$

$$\mathbf{G}(1) \propto \mathbf{I}_{3 \times 3}.$$

Note that the camera velocity cannot be constrained in the problem **(PC1)**. This means that velocity discontinuities can be observed at the beginning and the end of the task where the camera is motionless.

(PC2) Find $\mathbf{G}(t)$ minimizing

$$J_2 = \int_0^1 \dot{\mathbf{U}}^T \dot{\mathbf{U}} dt$$

subject to eq. (10), $\mathbf{v} = \dot{\mathbf{b}}$ and with boundary conditions:

$$\mathbf{G}(0) \propto \mathbf{G}_0$$

$$\mathbf{G}(1) \propto \mathbf{I}_{3 \times 3}$$

$$\mathbf{U}(0) = \mathbf{0}_{6 \times 1}$$

$$\mathbf{U}(1) = \mathbf{0}_{6 \times 1}.$$

In this case, the camera velocity is constrained to be $\mathbf{0}$ at the beginning and the end of the task. The boundary conditions are verified if $\mathbf{R}(0) = \mathbf{R}_0$, $\mathbf{b}(0) = \mathbf{b}_0$, $\mathbf{R}(1) = \mathbf{I}_{3 \times 3}$ and $\mathbf{b}(1) = \mathbf{0}$ (rigid motion problem). These boundary conditions are particularly important since they are the desired ones in the context of visual servoing.

The solutions of **PC1** and **PC2**, and a result about their dependence on camera calibration are given by the following proposition.

PROPOSITION 1. The optimal path of the collineation matrix in the sense of **PC1** and **PC2** is given by

$$\mathbf{G}(t) \propto (1 - q(t))\Phi_0 + (\mathbf{G}_0 - \Phi_0)\Gamma(\boldsymbol{\theta}_0, t) \quad (11)$$

where

$$\Gamma(\boldsymbol{\theta}_0, q(t)) = \mathbf{K} e^{[\boldsymbol{\theta}_0]_q(t)} \mathbf{K}^+ \quad \text{and} \quad \Phi_0 = \mathbf{K} \mathbf{b}_{0df} \mathbf{n}^f \mathbf{K}^+ \quad (12)$$

with $[\theta_0] = \log(\mathbf{R}_0^T)$, $\mathbf{b}_{0df} = \frac{\mathbf{b}_0}{df}$ and

$$\begin{cases} q(t) = t & \text{if PC1} \\ q(t) = -2t^3 + 3t^2 & \text{if PC2} \end{cases}.$$

The path given by eq. (11) is not affected by the error on intrinsic camera parameters if the collineation matrix at infinity at time 0 is known.

REMARK 1. The path given by Proposition 1 corresponds to a shortest distance path of the rotation matrix (minimal geodesic) with respect to an adequately chosen Riemannian metric on $SO(3)$ and to a straight line translation.

REMARK 2. As previously emphasized, the algorithm we used in practice to compute the collineation matrix at infinity at time 0 was designed for the case where the camera calibration is known. The independence of the computed trajectories with respect to the camera parameters can thus not be demonstrated in our experiments. However, we will see in Section 6 that there is no significant difference in the computed trajectories using a correct or a bad camera calibration. This means that the hypothesis of knowing the collineation matrix at infinity at time 0 has not a strong practical effect, and is more a technical point to demonstrate theoretically the independence property.

Proof. The solution of the rigid motion problem can be obtained by using standard optimal control results (or by following Park and Ravani 1997):

$$\mathbf{b}(t) = (1 - q(t))\mathbf{b}_0 \quad \text{and} \quad \mathbf{R}(t) = \mathbf{R}_0 \mathbf{e}^{[\theta_0]q(t)}. \quad (13)$$

According to eqs. (1) and (2), the corresponding collineation path is given by

$$\mathbf{G}(t) \propto \mathbf{K}(\mathbf{R}_0 \mathbf{e}^{[\theta_0]q(t)} + (1 - q(t))\mathbf{b}_{0df} \mathbf{n}^{fT}) \mathbf{K}^+.$$

This path is equivalent to the path given by

$$\begin{aligned} \mathbf{G}(t) \propto \mathbf{K} (\mathbf{R}_0 \mathbf{e}^{[\theta_0]q(t)} + (1 - q(t))\mathbf{b}_{0df} \mathbf{n}^{fT} \\ + \mathbf{b}_{0df} \mathbf{n}^{fT} \mathbf{e}^{[\theta_0]q(t)} - \mathbf{b}_{0df} \mathbf{n}^{fT} \mathbf{e}^{[\theta_0]q(t)}) \mathbf{K}^+ \end{aligned}$$

and can be rewritten as eq. (11) by denoting

$$\Gamma(\theta_0, q(t)) = \mathbf{K} \mathbf{e}^{[\theta_0]q(t)} \mathbf{K}^+ \quad \text{and} \quad \Phi_0 = \mathbf{K} \mathbf{b}_{0df} \mathbf{n}^{fT} \mathbf{K}^+.$$

We note that the matrix \mathbf{K} of camera internal parameters appears explicitly in the path given by eq. (11). However, the initial collineation matrix \mathbf{G}_0 is not affected by the error on intrinsic parameters since it is directly computed from image data. Furthermore, if $\hat{\mathbf{K}}$ is used instead of \mathbf{K} , we have

$$\hat{\Phi}_0 = \hat{\mathbf{K}} \hat{\mathbf{b}}_{0df} \hat{\mathbf{n}}^{fT} \hat{\mathbf{K}}^+ \quad \text{and} \quad \hat{\Gamma}(\theta_0, q(t)) = \hat{\mathbf{K}} \mathbf{e}^{[\hat{\theta}_0]q(t)} \hat{\mathbf{K}}^+.$$

By introducing eqs. (5) and (6) in $\hat{\Phi}_0$, we obtain

$$\begin{aligned} \hat{\Phi}_0 &= \hat{\mathbf{K}} \|\mathbf{n}^{fT} \delta \mathbf{K}^+\| \delta \mathbf{K} \mathbf{b}_{df} \frac{\mathbf{n}^{fT} \delta \mathbf{K}^+}{\|\mathbf{n}^{fT} \delta \mathbf{K}^+\|} \mathbf{K}^+ \\ &= \mathbf{K} \mathbf{b}_{0df} \mathbf{n}^{fT} \mathbf{K}^+ = \Phi_0. \end{aligned} \quad (14)$$

Furthermore (refer to eq. (8))

$$[\hat{\theta}_0] = \log(\hat{\mathbf{R}}_0^T) = \frac{\hat{\theta}_0}{2 \sin(\hat{\theta}_0)} (\hat{\mathbf{R}}_0^T - \hat{\mathbf{R}}_0) = \hat{\kappa} (\hat{\mathbf{R}}_0^T - \hat{\mathbf{R}}_0) \quad (15)$$

where $\hat{\kappa} = \frac{\hat{\theta}_0}{2 \sin \hat{\theta}_0}$ and $\hat{\theta}_0$ satisfies $1 + 2 \cos \hat{\theta}_0 = \text{Trace}(\hat{\mathbf{R}}_0)$. Since $\hat{\mathbf{R}}_0$ is similar to \mathbf{R}_0 (see eq. (4)), $\text{Trace}(\hat{\mathbf{R}}_0) = \text{Trace}(\mathbf{R}_0)$ that implies $\hat{\theta}_0 = \theta_0$ and thus $\hat{\kappa} = \kappa$. By inserting eq. (15) into eq. (7), we obtain

$$\begin{aligned} \hat{\Gamma}(\theta_0, q(t)) &= \hat{\mathbf{K}} \left(\mathbf{I} + \kappa \frac{\sin(\|\theta_0\|q(t))}{\|\theta_0\|} (\hat{\mathbf{R}}_0^T - \hat{\mathbf{R}}_0) \right. \\ &\quad \left. + \kappa^2 \frac{1 - \cos(\|\theta_0\|q(t))}{\|\theta_0\|^2} (\hat{\mathbf{R}}_0^T - \hat{\mathbf{R}}_0)^2 \right) \hat{\mathbf{K}}^+. \end{aligned}$$

Noticing that

$$\hat{\mathbf{K}} \hat{\mathbf{R}}_0 \hat{\mathbf{K}}^+ = \mathbf{K} \mathbf{R}_0 \mathbf{K}^+$$

we finally obtain

$$\hat{\Gamma}(\hat{\theta}_0, q(t)) = \Gamma(\theta_0, q(t)). \quad (16)$$

Thus, according to eqs. (14) and (16), the collineation trajectory given by eq. (11) is independent of the choice of the non-singular matrix $\hat{\mathbf{K}}$. Note that, in this proof, the collineation matrix at infinity at time 0 has been assumed known to derive eq. (14).

3.2. Two Interesting Particular Cases

In this section, we study the cases where the camera is displaced according to a pure rotational and a pure translational motion. We will see that in these cases the internal camera parameters do not appear at all.

3.2.1. Pure Rotational Motion

The collineation matrix is now a matrix similar to a rotation matrix (up to a scalar factor), that is

$$\mathbf{G} \propto \mathbf{K} \mathbf{R} \mathbf{K}^+. \quad (17)$$

The solutions of **PC1** and **PC2** and a nice property of the solutions are given by the following proposition.

PROPOSITION 2. The optimal path of the collineation matrix is given by

$$\mathbf{G}(t) \propto \mathbf{G}_0 \Gamma(\mathbf{g}_0, q(t)) \quad (18)$$

where $q(t) = t$ if **PC1**, $q(t) = -2t^3 + 3t^2$ if **PC2** and

$$\mathbf{\Gamma} = \mathbf{I} + \frac{\sin(\|\mathbf{g}_0\|q(t))}{\|\mathbf{g}_0\|} \{\mathbf{g}_0\} + \frac{1 - \cos(\|\mathbf{g}_0\|q(t))}{\|\mathbf{g}_0\|^2} \{\mathbf{g}_0\}^2 \quad (19)$$

with

$$\theta = \|\mathbf{g}_0\| = \arccos\left(\frac{1}{2}D_0^{-1/3}T_0 - 1\right) \quad (20)$$

$$\{\mathbf{g}_0\} = \frac{\|\mathbf{g}_0\|}{2 \sin \|\mathbf{g}_0\|} (D_0^{1/3}\mathbf{G}_0^+ - D_0^{-1/3}\mathbf{G}_0) \quad (21)$$

where $D_0 = \det(\mathbf{G}_0)$, $T_0 = \text{Trace}(\mathbf{G}_0)$.

The collineation path given by eq. (18) is independent of the internal camera parameters.

Proof. For a pure rotational motion $\Phi_0 = \mathbf{K}\mathbf{b}_{0df}\mathbf{n}^{fT}\mathbf{K}^+ = \mathbf{0}$, thus according to eq. (11):

$$\mathbf{G}(t) \propto \mathbf{G}_0\mathbf{\Gamma}(\theta_0, t).$$

Referring to eq. (7), $\mathbf{\Gamma}$ can be rewritten as follows:

$$\mathbf{\Gamma} = \mathbf{I} + \frac{\sin(\|\theta_0\|q(t))}{\|\theta_0\|} \mathbf{K}[\theta_0]\mathbf{K}^+ + \frac{1 - \cos(\|\theta_0\|q(t))}{\|\theta_0\|^2} \mathbf{K}[\theta_0]^2\mathbf{K}^+.$$

Using $\|\mathbf{g}_0\| = \|\theta_0\|$ and $\{\mathbf{g}_0\} = \mathbf{K}[\theta_0]\mathbf{K}^+$, the previous equation can be rewritten as eq. (19), and (see eq. (8))

$$\{\mathbf{g}_0\} = \mathbf{K}\log(\mathbf{R}_0^T)\mathbf{K}^+ = \frac{\|\mathbf{g}_0\|}{2 \sin \|\mathbf{g}_0\|} (\mathbf{K}\mathbf{R}_0^T\mathbf{K}^+ - \mathbf{K}\mathbf{R}_0\mathbf{K}^+). \quad (22)$$

Furthermore, from eq. (17), it is easy to show that

$$\mathbf{K}\mathbf{R}_0\mathbf{K}^+ = D_0^{-1/3}\mathbf{G}_0 \quad \text{and} \quad \text{Trace}(\mathbf{R}_0) = D_0^{-1/3}T \quad (23)$$

where $D_0 = \det(\mathbf{G}_0)$ and $T = \text{Trace}(\mathbf{G}_0)$. By introducing eq. (23) into eq. (22), we obtain eq. (21). Finally, using eqs. (9) and (23), we deduce eq. (20).

The initial value of the collineation \mathbf{G}_0 is not affected by errors on intrinsic parameters since it is extracted directly from image data. According to eqs. (18), (19), (20), and (21) and noticing that $\{\mathbf{g}_0\}$ and $\|\mathbf{g}_0\|$ depend only on \mathbf{G}_0 , the collineation path given by eq. (18) is independent of the camera parameters.

REMARK 2. As expected, the path given by Proposition 2 corresponds to a geodesic on $SO(3)$.

In the next subsection, the case of a pure translational camera motion is studied.

3.2.2. Pure Translational Motion

If the camera motion is a pure translation, the collineation matrix has the following particular form:

$$\mathbf{G} \propto \mathbf{I} + \mathbf{K}\mathbf{b}_{df}\mathbf{n}^{fT}\mathbf{K}^+. \quad (24)$$

PROPOSITION 3. The optimal path of the collineation matrix in the sense of **PC1** and **PC2** is given by

$$\mathbf{G}(t) \propto q(t)\mathbf{I} + \frac{(1 - q(t))}{\alpha_0} \mathbf{G}_0 \quad (25)$$

where $q(t) = t$ if **PC1**, $q(t) = -2t^3 + 3t^2$ if **PC2** and α_0 is a real solution of the equation

$$2\alpha^3 - T_0\alpha^2 + D_0 = 0. \quad (26)$$

The optimal smooth trajectories given by eq. (25) are not affected by error on intrinsic parameters.

Proof. For a pure translational motion, we have $\mathbf{\Gamma} = \mathbf{I}$, thus according to eqs. (11) and (24)

$$\mathbf{G}(t) \propto q(t)\mathbf{I} + (1 - q(t))(\mathbf{I} + \Phi_0)$$

and noticing that $\mathbf{G}_0 = \alpha_0(\mathbf{I} + \Phi_0)$, we deduce eq. (25). From eq. (24), we easily obtain

$$\begin{cases} \text{Trace}(\mathbf{G}_0) = T_0 = \alpha_0(3 + \mathbf{b}_{df}^T\mathbf{n}) \\ \text{Det}(\mathbf{G}_0) = D_0 = \alpha_0^3(1 + \mathbf{b}_{df}^T\mathbf{n}) \end{cases}$$

and by combining the previous equations, we deduce that α_0 is a solution of eq. (26).

The path of the collineation matrix given by eq. (25) is independent of the \mathbf{K} -matrix since the initial value of the collineation matrix \mathbf{G}_0 and thus α_0 (refer to eq. (26)) are independent of the camera intrinsic parameters.

REMARK 3. As expected also, the path given by eq. (25) corresponds to a straight line between the initial and desired camera positions.

4. Feature Trajectories in the Image

In this section, we first show how trajectories in the image space can be obtained from the collineation matrix path. Our approach is then illustrated by examples of image synthesis.

4.1. Image-Space Trajectories

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 $\mathbf{s}^*(1) = [x_1^*(1) \ y_1^*(1) \ \cdots \ x_n^*(1) \ y_n^*(1)]^T$. We denote by $\mathbf{x}_i^*(t) = [x_i^*(t) \ y_i^*(t) \ 1]^T$ the vector of homogeneous coordinates expressed in pixels of the projection of a 3D point

\mathcal{X}_i in the current desired image (at time t). We define vector $\mathbf{h}_i = \alpha_i(t)\mathbf{x}_i^*(t)$. It is well known that for all 3D points

$$\mathbf{h}_i(t) = \alpha_i(t)\mathbf{x}_i^*(t) = \mathbf{G}(t)\mathbf{x}_i^*(1) + \tau_i\boldsymbol{\gamma}(t) \quad (27)$$

where $\alpha_i(t)$ is a positive scaling factor depending on time, τ_i is a constant scaling factor null if the target point belongs to Π , and $\boldsymbol{\gamma}(t) = \mathbf{K}\mathbf{b}(t)$ represents the epipole in the current image (i.e., the projection in the image at time t of the optical center when the camera is in its desired position). After 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, $\boldsymbol{\gamma}(0) = \boldsymbol{\gamma}_0$, can also be computed directly from image data (i.e., $\boldsymbol{\gamma}_0$ is independent of the \mathbf{K} -matrix) (Faugeras 1993). Furthermore, it is easy to show (from eq. (13)) that the optimal trajectories of the epipole, with respect to the previously cited criteria, are of the form:

$$\boldsymbol{\gamma}(t) = (1 - q(t))\boldsymbol{\gamma}_0. \quad (28)$$

Such trajectories of the epipole are not affected by the error on intrinsic parameters since $\boldsymbol{\gamma}_0$ is directly computed from image data. Note also that the scaling factor τ_i is not time-dependent and can be computed directly from the initial and desired image data since (refer to eq. (27)):

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

We thus obtain

$$\tau_i = -\frac{(\mathbf{G}_0\mathbf{x}_i^*(1) \wedge \mathbf{x}_i^*(0))_1}{(\boldsymbol{\gamma}_0 \wedge \mathbf{x}_i^*(0))_1}.$$

where $(\mathbf{v})_j$ denotes the j th components of \mathbf{v} . To conclude, the vector \mathbf{h}_i is not affected by the error on intrinsic parameters since $\mathbf{G}(t)$, $\boldsymbol{\gamma}(t)$ and τ_i ($\forall i \in \{1 \dots n\}$) can be computed without error even if \mathbf{K} is unknown. The trajectories of the considered points in the image corresponding to an optimal camera path can thus also be computed without error, using

$$x_i^*(t) = \frac{(\mathbf{h}_i(t))_1}{(\mathbf{h}_i(t))_3} \quad y_i^*(t) = \frac{(\mathbf{h}_i(t))_2}{(\mathbf{h}_i(t))_3}. \quad (29)$$

4.2. Example: Synthesis of Intermediate Views

In this section, our approach is illustrated by two experiments. Experiments were performed with images acquired using a CCD camera mounted on the effector of a 6-DoF manipulator. Intermediate views between the initial and the desired images and corresponding to the minimum energy (Figure 2) and acceleration problems (Figure 3) are constructed. The camera displacement between initial and desired images (boxed in Figures 2 and 3) is very large. The initial collineation matrix is:

$$\mathbf{G}_0 = \begin{bmatrix} 0.4603 & 0.4145 & -597.3284 \\ -0.2476 & 0.6107 & -376.0835 \\ -0.0001 & 0.0006 & -1.5281 \end{bmatrix}.$$

Intermediate images synthesized using eq. (29) and the solution of the problems **PC1** and **PC2** are given in Figures 2 and 3, respectively. The corresponding camera trajectories are plotted in Figures 4(a) and (b). Note that only the temporal behavior is different for the two experiments.

5. Application to Visual Servoing

To track the trajectories using an image-based control scheme, we use the task function approach introduced in Samson, Espiau, and Le Borgne (1991). A vision-based task function \mathbf{e} to be regulated to $\mathbf{0}$ is defined by (Espiau, Chaumette, and Rives 1992)

$$\mathbf{e} = \hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}^*(t)). \quad (30)$$

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 $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ (λ being a proportional gain). The corresponding control law is

$$\mathbf{T}_c = -\lambda\mathbf{e} - \frac{\partial\mathbf{e}}{\partial t} \quad (31)$$

where $\frac{\partial\mathbf{e}}{\partial t}$ is an approximation of $\frac{\partial\mathbf{e}}{\partial t}$ involved to minimize the tracking error. Using such a control law, a well-known sufficient condition to ensure global asymptotic stability of the system is (Samson, Espiau, and Le Borgne 1991)

$$\hat{\mathbf{L}}^+\mathbf{L} > 0. \quad (32)$$

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

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

where $\mathbf{m} = [m_x \ m_y \ 1]^T = \mathbf{K}^+\mathbf{x}$. When \mathbf{s} is composed of the image coordinates \mathbf{x}_i of n points, the corresponding interaction matrix is

$$\mathbf{L}(\mathbf{s}, \mathbf{Z}) = [\mathbf{L}^T(\mathbf{x}_1, Z_1) \dots \mathbf{L}^T(\mathbf{x}_n, Z_n)]^T. \quad (33)$$

A classical choice for $\hat{\mathbf{L}}$ is $\mathbf{L}(\mathbf{s}(1), \hat{\mathbf{Z}}(1))$ (i.e., the value of \mathbf{L} at the final desired position). In this case, condition (32) is ensured only in a neighborhood of the desired position (Chaumette 1998). We will use the value of \mathbf{L} at the current desired position for $\hat{\mathbf{L}}$ (i.e., $\hat{\mathbf{L}} = \mathbf{L}(\mathbf{s}^*(t), \hat{\mathbf{Z}}^*(t))$ rather

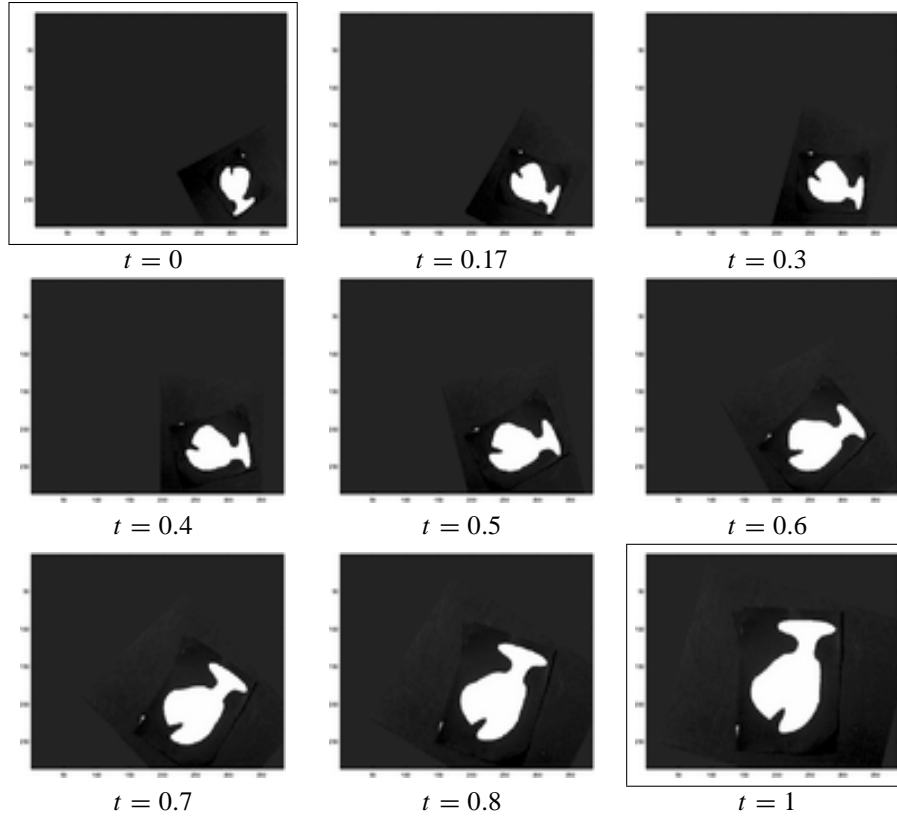


Fig. 2. Image trajectories: minimum energy.

than $\mathbf{L}(\mathbf{s}(1), \hat{\mathbf{Z}}(1))$. With this choice, condition (32) is ensured along the planned trajectories and not only in a neighborhood of the final desired position.

The interaction matrix depends on the $\hat{\mathbf{Z}}^*$ -vector. According to eq. (3), this vector can be rewritten as $\hat{\mathbf{Z}}^*(t) = \hat{d}^f \boldsymbol{\Sigma}^*(t)$ where \hat{d}^f is an estimated value of d^f (i.e., the only parameter that has to be introduced “by hand”) and $\boldsymbol{\Sigma}^*(t) = [\rho_1^*(t) \cdots \rho_n^*(t)]$. The ratio $\rho_i^*(t)$ can be estimated (when $\hat{\mathbf{K}}$ is used instead of \mathbf{K}) using relations (3):

$$\left\{ \begin{array}{ll} \hat{\rho}_i^*(t) = \frac{\hat{r}^*(t)}{\hat{\mathbf{n}}^{*T}(t) \hat{\mathbf{m}}_i(t)} & \text{if } \mathcal{X}_i \in \Pi \\ \hat{\rho}_i^*(t) = \hat{\beta} \frac{\|\hat{\mathbf{b}}_{d^f}\|}{\|\hat{\beta} \hat{\mathbf{m}}_i - \hat{\mathbf{R}}\|} & \text{if } \mathcal{X}_i \notin \Pi \end{array} \right. \quad (34)$$

The previous relations can be estimated using only the analytical form of the parameters obtained during the path planning step, with (Mezouar 2001)

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

where

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

In practice, we use these last relations to compute $\rho_i^*(t)$ since the analytical forms of all these components are known.

If the target is known to be motionless, we get from eq. (30):

$$\frac{\partial \mathbf{e}}{\partial t} = -\hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t}$$

and the control law (31) can be rewritten as follows

$$\mathbf{T}_c = -\lambda \mathbf{e} + \hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t} \quad (35)$$

where the term $\hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t} = \hat{\mathbf{L}}^+ \left[\frac{\partial x_1^*}{\partial t} \frac{\partial y_1^*}{\partial t} \cdots \frac{\partial x_n^*}{\partial t} \frac{\partial y_n^*}{\partial t} \right]^T$ allows us to compensate the tracking error. More precisely, we have from eq. (27)

$$\frac{\partial \mathbf{x}_i^*}{\partial t} = \frac{1}{\alpha_i(t)} \left[\frac{\partial \mathbf{G}}{\partial t} \mathbf{x}_i^*(1) + \beta_i \frac{\partial \boldsymbol{\gamma}}{\partial t} - \frac{\partial \alpha_i}{\partial t} \mathbf{x}_i^*(t) \right]. \quad (36)$$

If we rewrite the collineation and the epipole as follows

$$\mathbf{G}(t) = \begin{bmatrix} \mathbf{G}_1(t) \\ \mathbf{G}_2(t) \\ \mathbf{G}_3(t) \end{bmatrix} \quad \boldsymbol{\gamma}(t) = \begin{bmatrix} \gamma_1(t) \\ \gamma_2(t) \\ \gamma_3(t) \end{bmatrix}$$

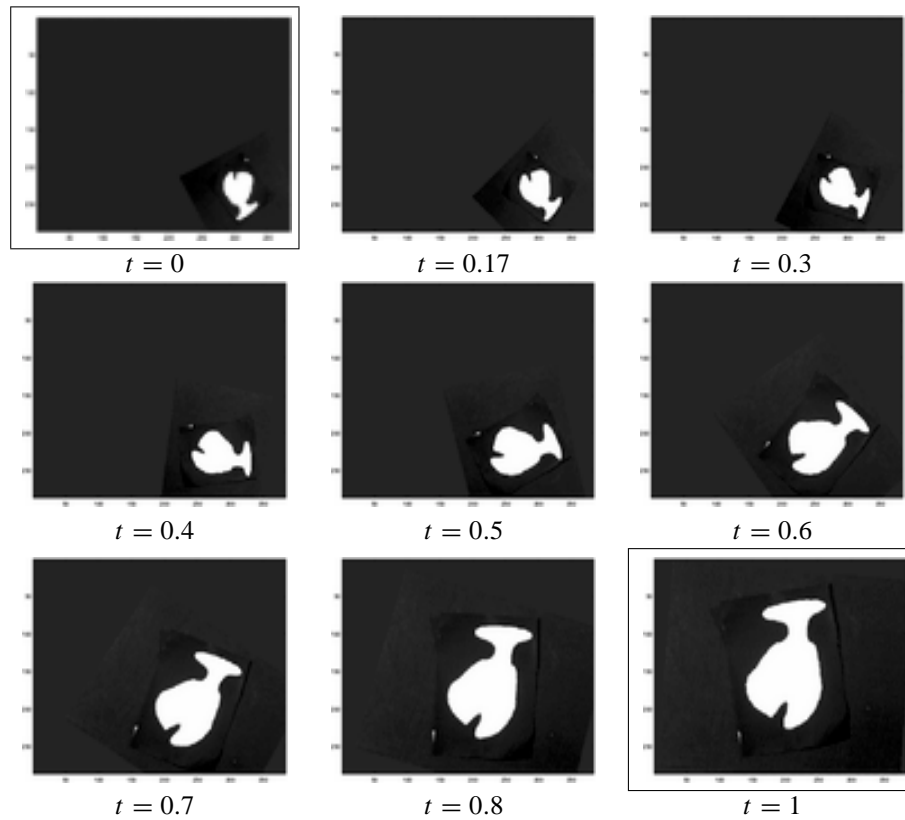


Fig. 3. Image trajectories: minimum acceleration.

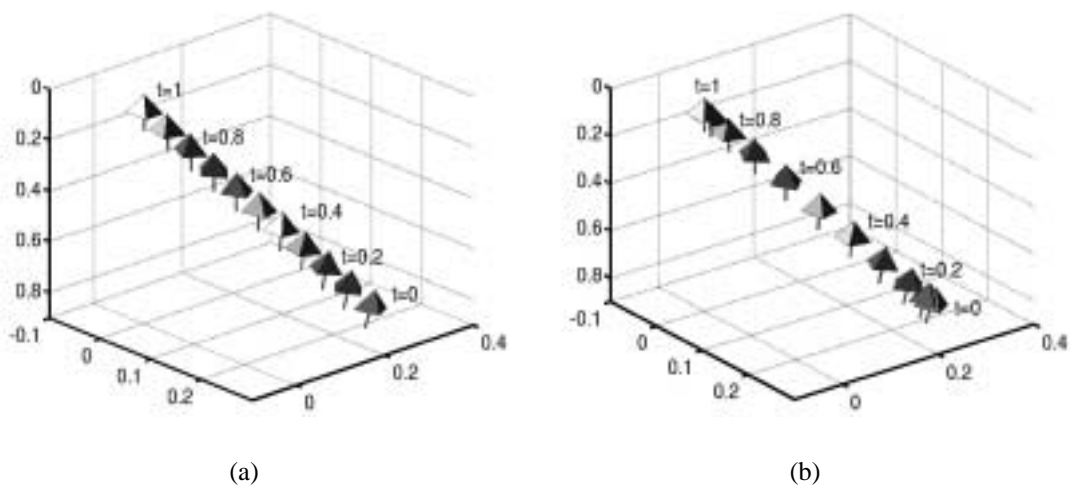


Fig. 4. Camera trajectory: (a) minimum energy and (b) minimum acceleration.

we obtain from eqs. (11), (27), and (28):

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{G}}{\partial t} = \dot{q}(t) [-\Phi_0 + (\mathbf{G}_0 + \Phi_0)\Gamma(t)] = \begin{bmatrix} \frac{\partial \mathbf{G}_1}{\partial t} \\ \frac{\partial \mathbf{G}_2}{\partial t} \\ \frac{\partial \mathbf{G}_3}{\partial t} \end{bmatrix} \\ \frac{\partial \gamma}{\partial t} = -\dot{q}(t)\gamma_0 = \begin{bmatrix} \frac{\partial \gamma_1}{\partial t} \\ \frac{\partial \gamma_2}{\partial t} \\ \frac{\partial \gamma_3}{\partial t} \end{bmatrix} \\ \alpha_i(t) = \mathbf{G}_3(t)\mathbf{x}_i^*(1) + \tau_i\gamma_3(t) \\ \frac{\partial \alpha_i}{\partial t} = \frac{\partial \mathbf{G}_3}{\partial t}\mathbf{x}_i^*(1) + \tau_i\frac{\partial \gamma_3}{\partial t} \end{array} \right. .$$

The term $\frac{\partial \mathbf{s}^*}{\partial t}$ is finally obtained by introducing the previous relations in eq. (36).

6. Experimental Results

The proposed method has been tested in a positioning task with respect to an unknown scene. We used a CCD camera mounted on a 6-DoF manipulator. We have used for the experiments two targets. The first target is composed of nine white marks. The extracted visual features are the image coordinates of the center of gravity of each mark. The second target is a complex image. The extracted features are the image coordinates of interesting points. The desired images have been acquired during an off-line step. The algorithm proposed in Malis and Chaumette (2000) has been used to obtain the initial collineation matrix from which the initial collineation matrix at infinity is extracted using the algorithm proposed in Faugeras and Lustman (1988).

6.1. First Target

The target is composed of nine white marks lying on three different planes (see Figure 8). The images corresponding to the desired and initial camera positions are given in Figures 5(a) and (b), respectively. As can be seen in Table 1, the corresponding camera displacement is very large. In order to check the robustness of the proposed approach with respect to modeling errors, two different sets of parameters have been used:

1. **correct calibration**—the correct intrinsic parameters and the correct value of d^f (i.e., 50 cm) have been used (see Figures 6 and 8);
2. **bad calibration**—an error of 50% has been added on the intrinsic parameters, while d^f has been set to 100 cm (see Figures 7 and 9).

Figures 6 and 7 present the results using the minimal energy criterion and Figures 8 and 9 give the results using the minimal acceleration criterion. In the following, only the results related to the minimal energy criterion are commented on, since the

comments are also valid for the minimal acceleration problem. However, let us note that the temporal behavior depends on the chosen cost function, as can be seen for example in the figures representing the camera velocity.

1. **Correct calibration.** Planned and tracked trajectories are plotted in Figures 6(a) and (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 6(e). It confirms the previous comment since the maximal error remains small (always less than five 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 6(d). The convergence of the coordinates to their desired value demonstrates the correct realization of the task. The computed camera velocity is given in Figure 6(f). Note finally that the camera optical center moves along a straight line as can be seen in Figure 6(c).
2. **Bad calibration.** First, we note that the planned trajectories obtained with or without modeling errors are similar (refer to Figures 6(a) and 7(a)). This confirms the robustness of the path planning process with respect to camera calibration errors. Secondly, as can be seen in Figures 7(a) and (b), the planned and followed trajectories in the bad calibration case are also similar. The image-based control scheme is indeed particularly robust with respect to modeling errors when the error function to regulate is small (the tracking errors, given in Figure 7(e), remain small during the servoing, i.e., less than eight pixels). The camera velocity is plotted in Figure 7(f). Once again, the task is correctly realized. This is shown by the convergence of the image point coordinates to their desired value (refer to Figure 7(d)).

6.2. Second Target

The extracted visual features are now the image coordinates of interesting points obtained using the Harris detector (Harris and Stephens 1988). These points are matched using the algorithm described in Zhang et al. (1995). These points are represented by crosses in the initial and desired images (refer to Figure 10). The presented results have been obtained using the minimum energy cost function. In order to check the robustness with respect to modeling errors, we carried out two experiments. In the first experiment, the camera parameters given by the camera manufacturer and a correct approximation of the depth d^f (i.e., $d^f = 0.9$ m) have been used (see Figure 11 and Extension 1). In the second experiment, an error of 30% has been added to the camera parameters, while d^f has been set to 0.5 m (Figure 12 and Extension 2). As can be seen in Figures 11(a), 11(b), 12(a) and 12(b), the planned and the tracked trajectories are similar. The trajectories obtained

Table 1. Camera Displacement

Translation (mm)	$t_x = -195$	$t_y = -610$	$t_z = -1455$
Rotation (deg)	$(u\theta)_x = -68$	$(u\theta)_y = -41$	$(u\theta)_z = -144$

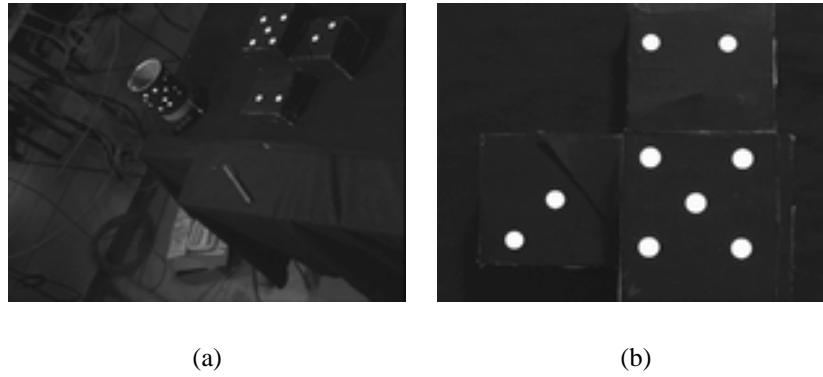
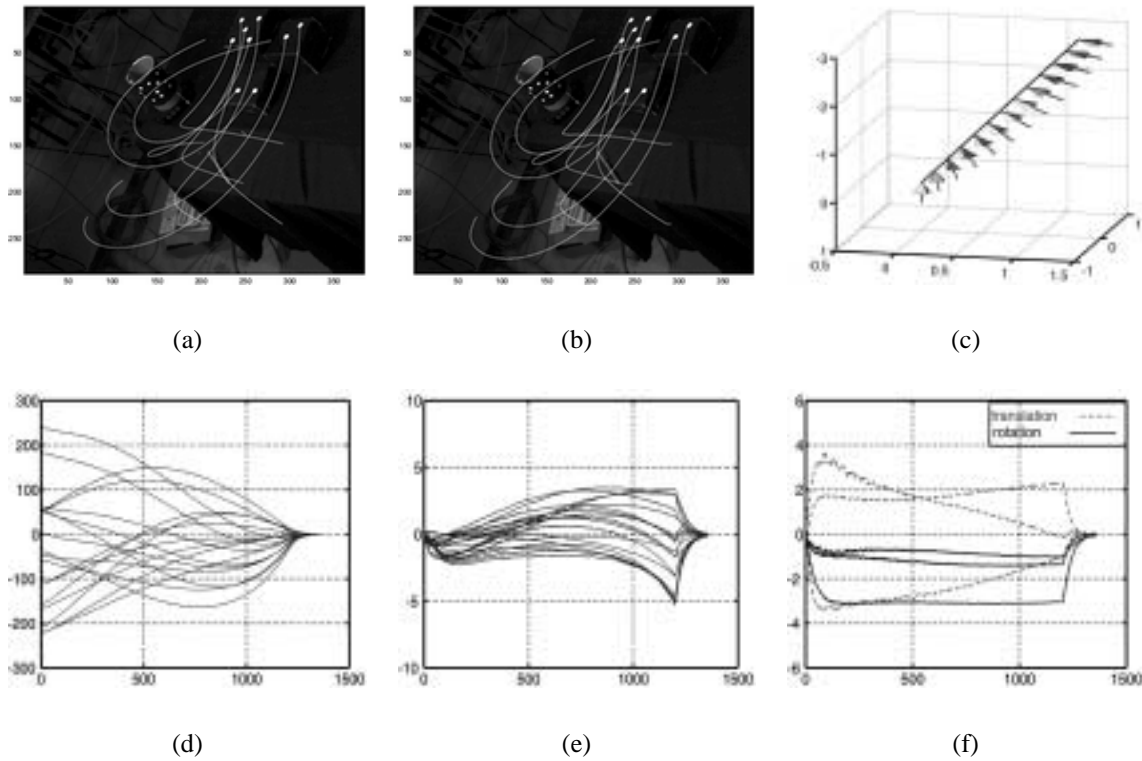


Fig. 5. First target: (a) initial image and (b) desired image.


 Fig. 6. First target, minimum energy criterion and correct calibration: (a) planned trajectories; (b) followed trajectories; (c) camera trajectory; (d) error in image point coordinates (pixels); (e) tracking error (pixels); (f) velocities (cm s^{-1} and deg s^{-1}).

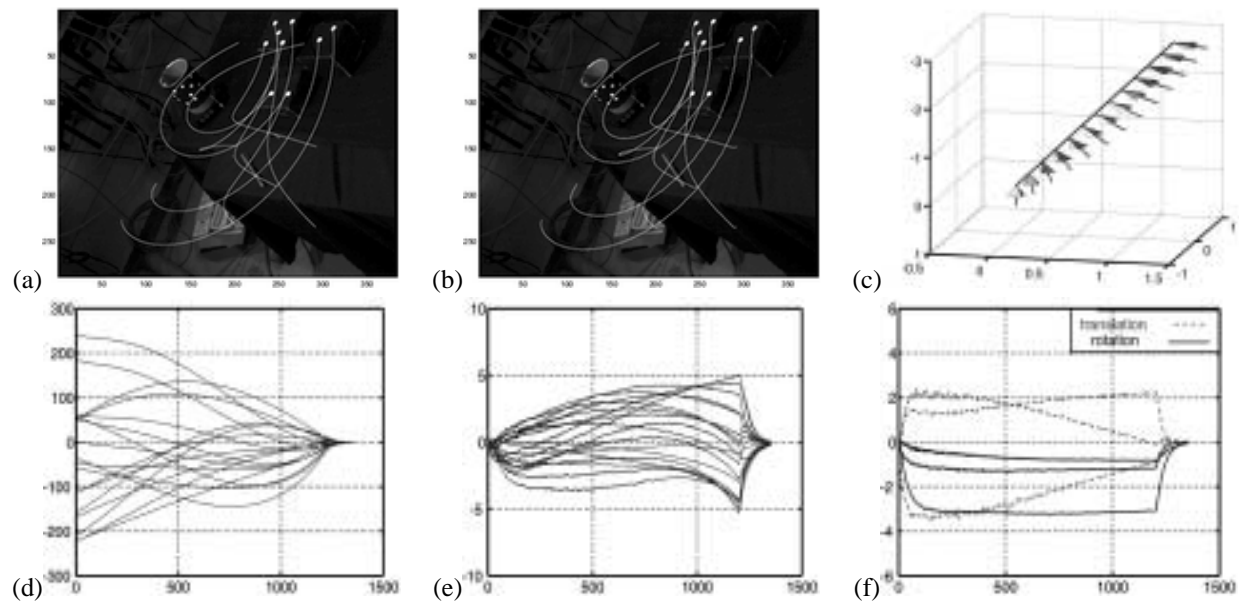


Fig. 7. First target, minimum energy criterion and bad calibration: (a) planned trajectories; (b) followed trajectories; (c) camera trajectory; (d) error in image point coordinates (pixels); (e) tracking error (pixels); (f) velocities (cm s^{-1} and deg s^{-1}).

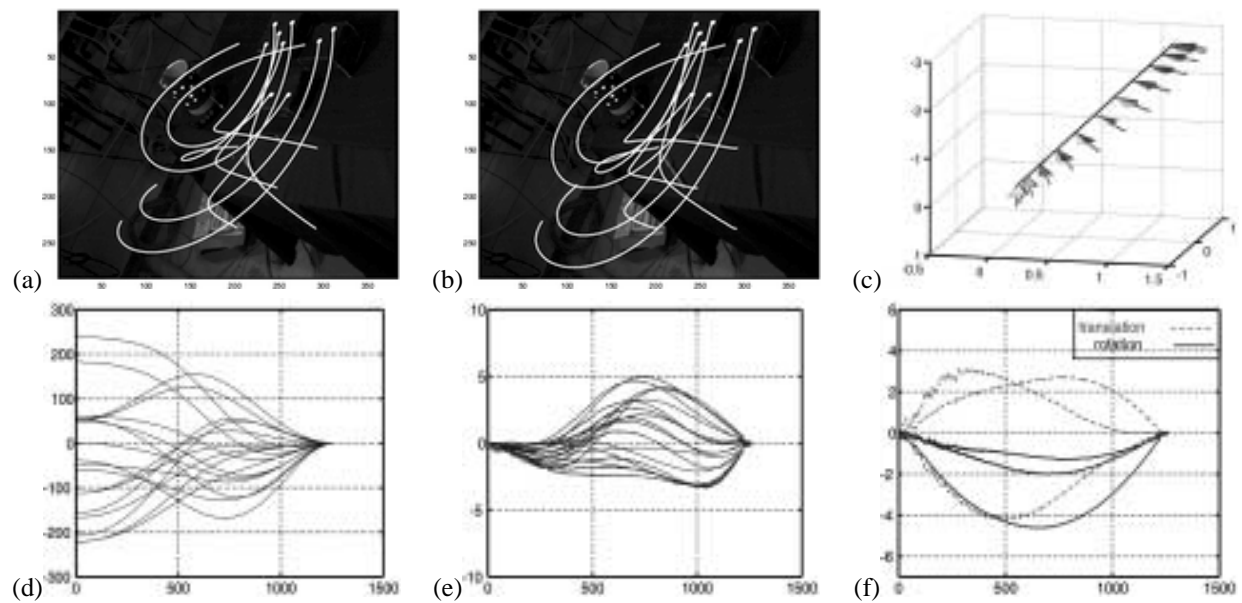


Fig. 8. First target, minimum acceleration criterion and correct calibration: (a) planned trajectories; (b) followed trajectories; (c) camera trajectory; (d) error in image point coordinates (pixels); (e) tracking error (pixels); (f) velocities (cm s^{-1} and deg s^{-1}).

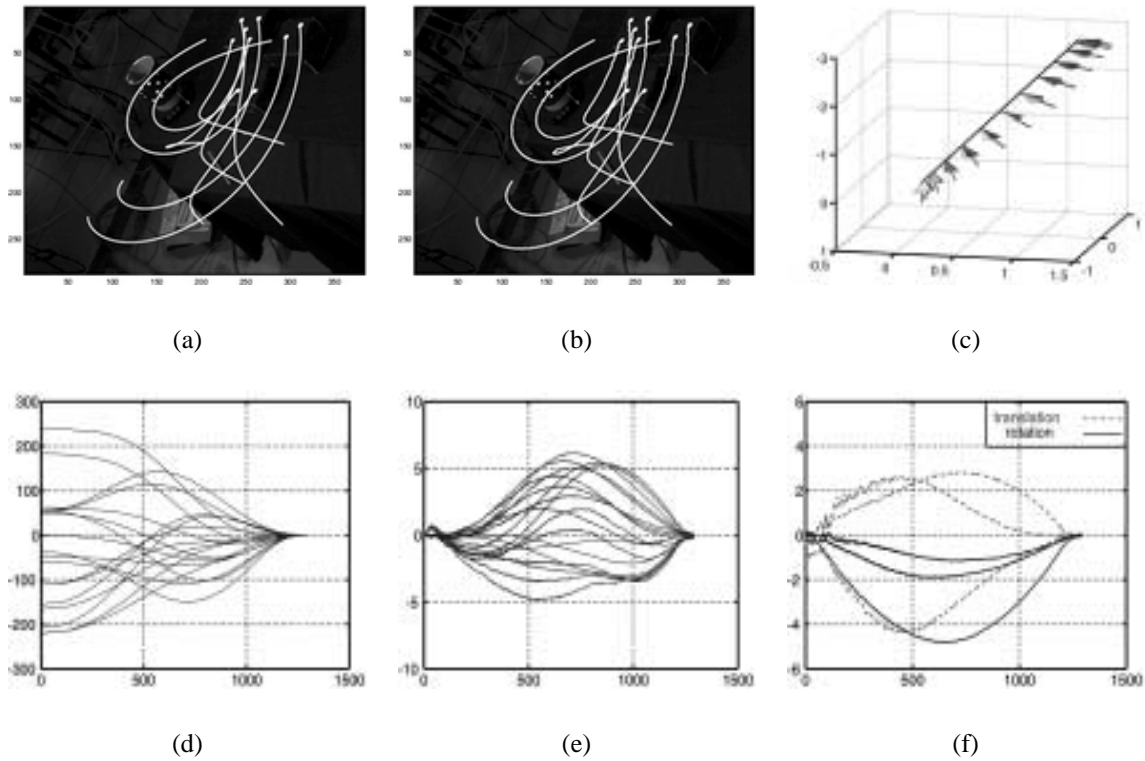


Fig. 9. First target, minimum acceleration criterion and bad calibration: (a) planned trajectories; (b) followed trajectories; (c) camera trajectory; (d) error in image point coordinates (pixels); (e) tracking error (pixels); (f) velocities (cm s^{-1} and deg s^{-1}).

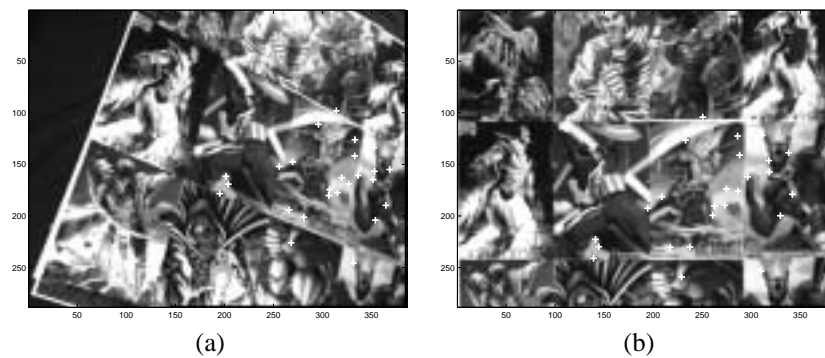


Fig. 10. Second target: (a) initial image and (b) desired image.

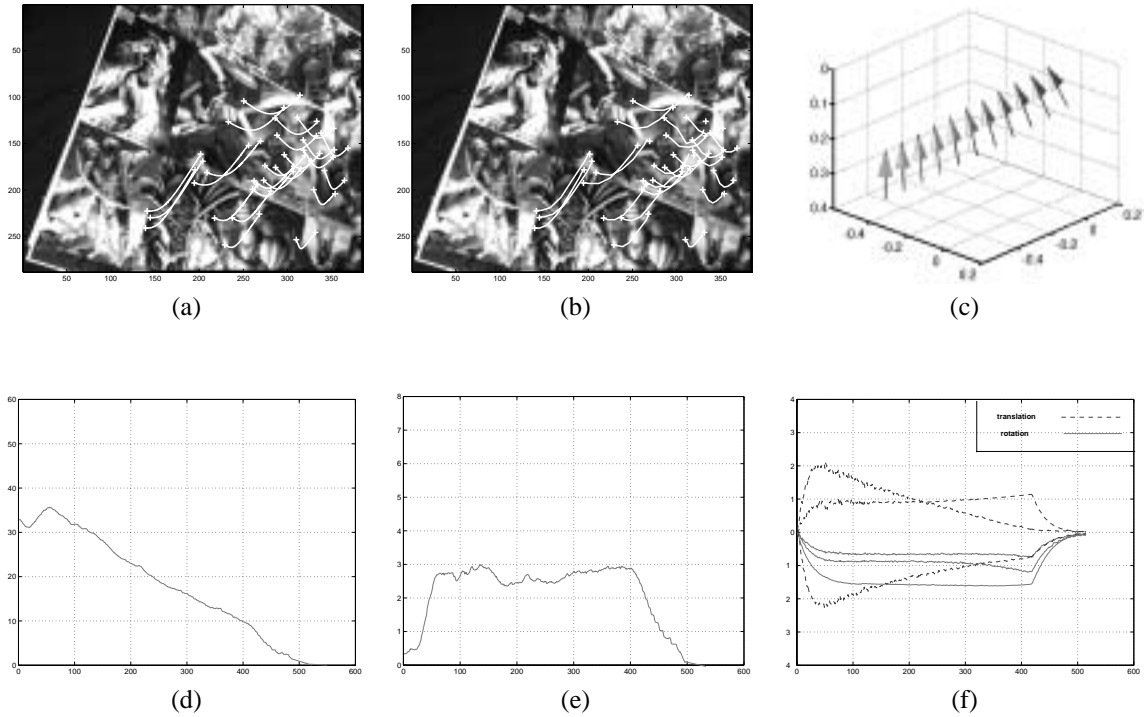


Fig. 11. Second target using correct calibration: (a) planned trajectories; (b) followed trajectories; (c) camera trajectory; (d) error in image point coordinates (pixels); (e) tracking error (pixels); (f) velocities (cm s^{-1} and deg s^{-1}).

with correct parameters are close to those obtained with bad parameters (compare Figure 11(a) with Figure 12(a) and Figure 11(b) with Figure 12(b)). This confirms, once again, the robustness of the path planning and of the control scheme with respect to calibration errors and errors on d^f . 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 Figures 11(e) and 12(e) remains small in both cases (less than four pixels). The convergence to 0 of the error on the coordinates between the current and the desired images (defined as $\frac{1}{2n} \sum_1^n (x_i(t) - x_i^*(1)) + (y_i(t) - y_i^*(1))$) and plotted in Figures 11(d) and 12(d) demonstrates the correct realization of the positioning task in both cases. As can be seen in Figures 11(c) and 12(c), the camera optical center moves along a straight line.

7. $(N + 1)$ Images Framework

Classical visual servoing techniques make assumptions on the link between the initial and desired images, limiting the applicability of these techniques to relatively small displacement when the scene is complex. Indeed, if a sufficient number of image features cannot be matched in these images, the visual servoing cannot be realized (see, for example, the initial and

desired images given in Figures 13(a) and (b)). A possible solution to cope with this deficiency is to use relay images acquired off-line. In the next section, we show how our scheme can be extended to this framework.

7.1. Images Trajectories

Assume now that a set of $N + 1$ relay images $\mathcal{I} = \{I_0 \cdots I_N\}$ is available and that some image features can be extracted and matched between two successive images. 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 eqs. (1) and (2)) between images I_i and I_N can easily be obtained noticing that

$$\mathbf{G}_{i,N} = \mathbf{G}_i = \mathbf{G}_{i,i+1} \mathbf{G}_{i+1,i+2} \cdots \mathbf{G}_{N-1,N}. \quad (37)$$

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 with a set of $N + 1$ time parameters $\{t_0 \cdots t_{N-1}, t_N\}$, we want to determine a continuous and piecewise differentiable matrix function $\mathbf{G}(t)$ such that $\mathbf{G}(t_i) = \mathbf{G}_i$ for $i \in \{0 \cdots N\}$ and such that $\mathbf{G}(t)$ corresponds to a minimal length camera trajectory. This problem

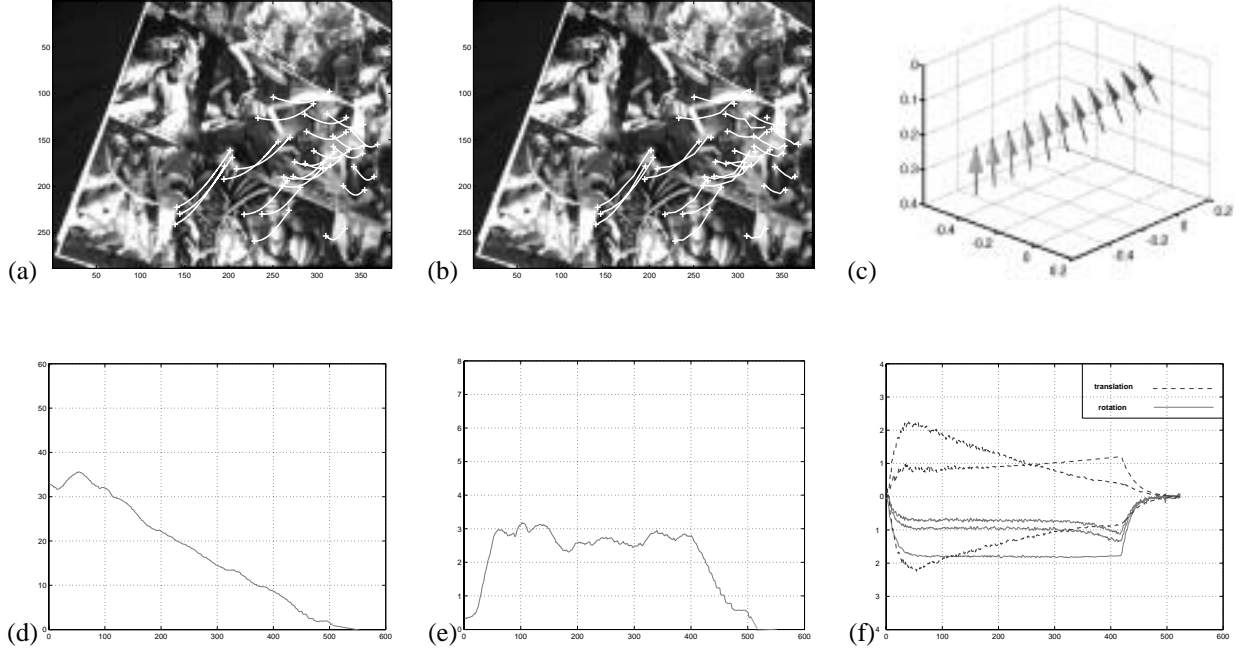


Fig. 12. Second target using bad calibration: (a) planned trajectories; (b) followed trajectories; (c) camera trajectory; (d) error in image point coordinates (pixels); (e) tracking error (pixels); (f) velocities (cm s⁻¹ and deg s⁻¹).

can be formulated as follows (problem **PM**)

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

for $i = 0 \dots N$ and with $\mathbf{U} = [\mathbf{v}^T \ \boldsymbol{\omega}^T]^T$ where $\boldsymbol{\omega}$ is defined by eq. (10), $\mathbf{v} = \dot{\mathbf{b}}$, 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 problem **PM** can be obtained in the same way as the solution of **PC1** (Mezouar 2001)

$$\mathbf{G}(\tau) \propto (1 - \tau)\Phi_{i-1} + \tau\Phi_i + (\mathbf{G}_{i-1} - \Phi_{i-1})\Gamma \quad (38)$$

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

$$\Gamma(\theta_i, \tau) = \mathbf{K}e^{[\theta_i]_{\times} \tau} \mathbf{K}^+, \quad \Phi_i = \mathbf{K} \mathbf{b}_{dfi} \mathbf{n}^{fT} \mathbf{K}^+ \quad (39)$$

with $[\theta_i]_{\times} = \log(\mathbf{R}_{i-1}^T \mathbf{R}_i)$. By introducing eqs. (4), (5), and (6) in eq. (38), it can be shown that the path given by eq. (38) is not affected by the error on camera intrinsic parameters (the proof is similar to the case of two images). The features are interpolated as for the case of two images (Section 4) and the displacement is then carried out using the control scheme described in Section 5. Note that recent work in image database analysis to solve the problem of retrieving and delivering images from large database using query (De Marsico, Cinque, and Levialdi 1997) can be exploited to extract the set of relay images from an image database. Our scheme to obtain the relay images can be summarized as follows (Mezouar et al. 2002).

1. In a first off-line step, the camera 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.
2. Points of interest of all these images are extracted using the Harris detector (Harris and Stephens 1988), and some invariants are computed (Schmid and Mohr 1997). This allows us, on the one hand, to index these images in a database which will be used to quickly retrieve images acquired during the operational phase, and on the other hand to match images by pair.
3. 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 can be matched. The edges are valuated in a way inversely proportional to the number of matched image features; the matching is realized using the algorithm proposed in Zhang et al. (1995).
4. In the 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 algorithm. We thus obtain an ordered set of

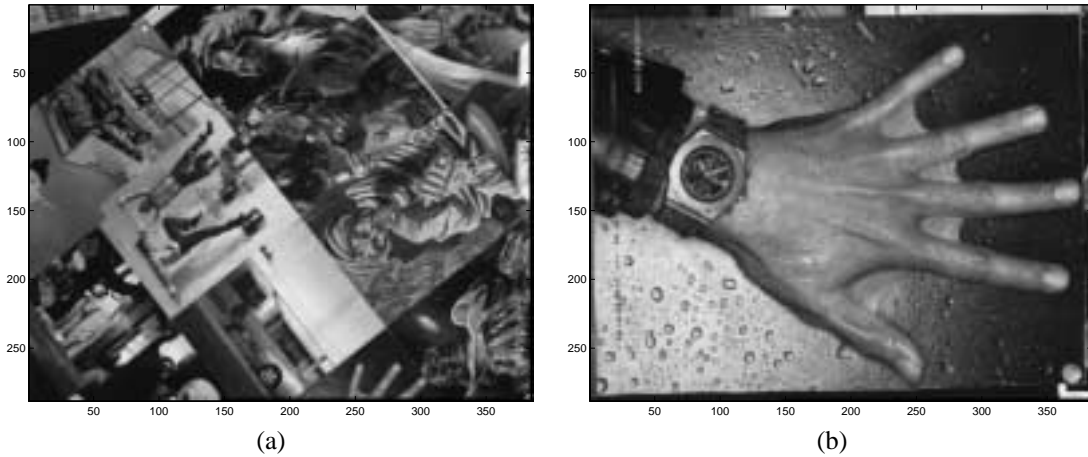


Fig. 13. (a) Initial image and (b) desired image.

relay images such that, between two successive images of this set, a sufficient number of image features can be matched.

7.2. 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 Figures 13(a) and (b). In this case, the SFMJF is impossible to realize. However, from the graph built with the image database, eight relay images are obtained (see Figure 14). The trajectories of interesting points are then planned. The planned and followed trajectories are given in Figures 14 and 15. We note that these trajectories are similar. The camera trajectory is given in Figure 16(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 Figure 16(b)) remains sufficiently small (always less than five pixels) to ensure a good behavior of the control scheme. Extension 3 (refer to the Appendix for details) gives an internal view (i.e., a view by the controlled camera) of the positioning task.

8. Potential-Based Path Planning

The path planning approach described in the previous sections, coupled to a purely image-based control, enables us to realize large robot displacements with a strong robustness with respect to modeling errors. Additionally, closed-form image trajectories have been obtained, hence only low computational cost is required for on-line implementation. However, the limitations of the stable region of image-based control and the non-optimality or non-validity of the induced camera trajectory were not the only reasons for image-based control failures. For example, the features of interest may leave the camera field of view and the robot may reach its mechanical

limits. To ensure the success of the task whatever the displacement to realize, we have tried to embed in a constrained global optimization process the previously described path planning scheme (Mezouar 2001). However, the global optimization process requires a high computational time and an initialization step. It is thus not suitable for on-line implementation. To cope with this problem, a second path planning scheme based on a local optimization of potential function (Khatib 1986; Latombe 1991) is now briefly presented. In this scheme, constraints (visibility and joint limits avoidance) can easily be introduced at the path planning level.

8.1. Principle

In this approach, the robot motions are under the influence of an artificial potential field (V) defined as the sum of an attractive potential (V_a) pulling the robot towards the goal configuration (Υ_*) and a repulsive potential (V_r) pushing the robot away from the obstacles. Motion planning is performed in an iterative fashion. At each iteration, an artificial force $\mathbf{F}(\Upsilon)$, where the 6×1 vector Υ represents a parametrization of the robot workspace $\mathcal{W} \subset \mathbf{R}^p$, is induced by the potential function. This force is defined as $\mathbf{F}(\Upsilon) = -\tilde{\nabla}_{\Upsilon}^T V$ where $\tilde{\nabla}_{\Upsilon}^T V$ denotes the transpose of the gradient vector of V at Υ . Using these conventions, $\mathbf{F}(\Upsilon)$ can be decomposed as the sum of two vectors, $\mathbf{F}_a(\Upsilon) = -\tilde{\nabla}_{\Upsilon}^T V_a$ and $\mathbf{F}_r(\Upsilon) = -\tilde{\nabla}_{\Upsilon}^T V_r$, which are respectively called the attractive and repulsive forces. Path generation proceeds along the direction of $\mathbf{F}(\Upsilon)$ regarded as the most promising direction of motion. Thus, each segment is oriented along the negated gradient of the potential function computed at the configuration attained by the previous segment. The discrete-time trajectory is given by the transition equation

$$\Upsilon_{k+1} = \Upsilon_k + \varepsilon_k \frac{\mathbf{F}(\Upsilon_k)}{\|\mathbf{F}(\Upsilon_k)\|} \quad (40)$$

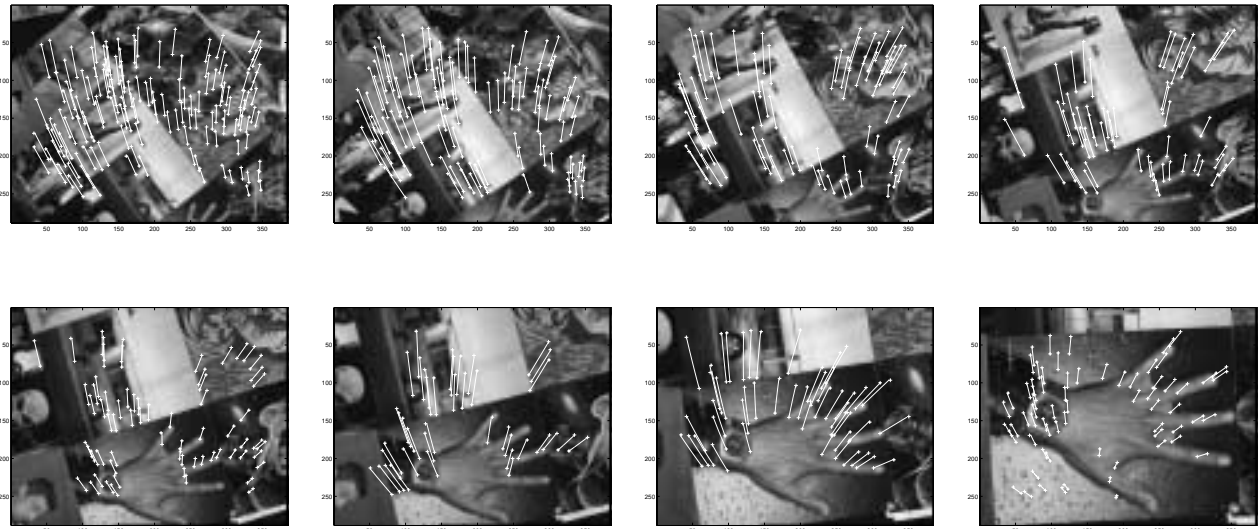


Fig. 14. Planned trajectories.

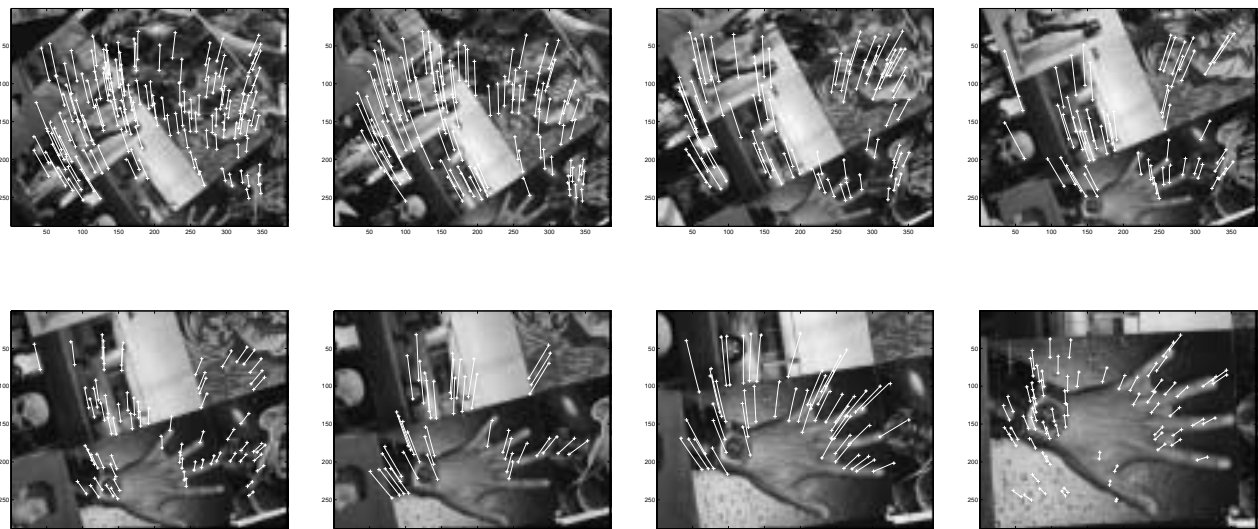


Fig. 15. Followed trajectories.

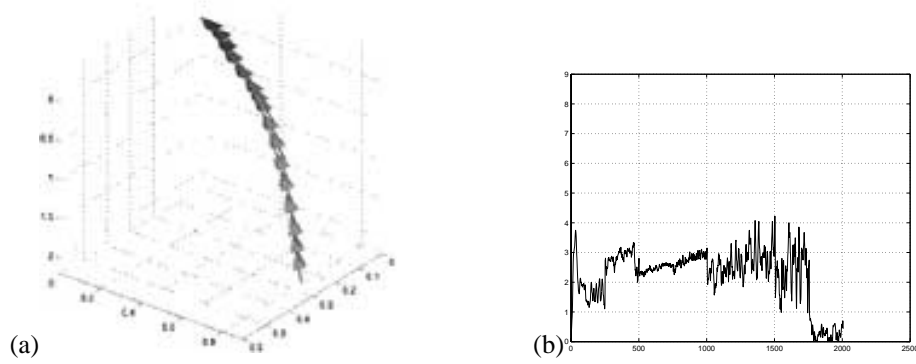


Fig. 16. (a) Camera trajectory and (b) tracking error (pixels).

where k is the increment index and ε_k is a positive scaling factor denoting the length of the k th increment. In our case, the control objective can be formulated as follows. To transfer the system to a desired point in the sensor space satisfying the following constraints:

1. the image trajectories correspond to a valid robot trajectory;
2. all the considered image features remain in the camera field of view;
3. the robot joints remain between their limits.

To deal with the first constraint, the motion is first (partially) planned in the 3D Cartesian space using the transition equation (40) and then projected in the image space. Only the rotation matrix and scaled translation vector trajectories are necessary to obtain the corresponding features trajectories in the image space. This property allows us to compute the image trajectories without knowing the CAD model of the object (model-free method), using eqs. (1), (2), and (27).

The attractive potential (V_r) pulling the robot toward the goal configuration (Υ_*) is defined in the 3D Cartesian space as a parabolic function reaching its minimum at Υ_* . The second and the third constraints are introduced through a repulsive potential V_s defined in the image and a repulsive potential V_q defined in the joint space. One way to create a potential barrier around the camera field of view is to define V_s as an increasing function of the distance between the object projection and the image limits (see Figure 17(a)). In the same way, a potential barrier can be created around the robot joint limits (see Figure 17(b)). A general description of such functions and the derivation of the associated artificial forces are given in Mezouar and Chaumette (2002). The total force is given by

$$\mathbf{F} = \mathbf{F}_r + \gamma \mathbf{F}_s + \chi \mathbf{F}_q \quad (41)$$

where the scaling factors γ and χ allow us to adjust the relative influence of the different forces. Discretized trajectories

in the image space are then computed. In order to design continuous and differentiable curves and to improve the dynamic behavior of the system, the discretized trajectories are interpolated using cubic B-spline functions. The continuous trajectories are then integrated in an image-based control loop in a similar way than for the first planning scheme.

As for the previous path planning scheme, it can be shown that the potential-based scheme is particularly robust with respect to calibration errors (Mezouar and Chaumette 2002). Contrarily to the previous method, constraints can be integrated in an easy way. In counterpart, the analytical form of the trajectories in the image space are not known and the camera trajectory is deviated from the optimal one when repulsive forces are involved. Finally, let us note that the total force given by (41) may potentially lead to local minima. This is intrinsically due to the local and incremental strategy of the potential field approach. In practice, we have never encountered configurations leading to local minima and the parameters γ and χ have been fixed to 1. A basic strategy to take out of potential local minima would be to execute a motion favoring the repulsive force (i.e., by increasing parameters γ and χ). Obviously such strategy makes no formal guarantee to reach the global minimum, and in some cases, there is no solution to satisfy all the constraints. From an intuitive point of view and following the reasoning proposed in Chesi et al. (2002), the repulsive force related to the visibility constraint will induce a backward motion if no other motion is possible (which is the case when at least two image points leave the field of view on opposite image limits simultaneously). As soon as the joints limits are taken into account, it may be impossible to realize such backward motion, leading thus to the failure of the path planning. In Cowan and Koditschek (1999), potential functions free of stable local minima (called *navigation functions* (Rimon and Koditschek 1992)) are constructed to guarantee the global stability of the system. However, constructing such a navigation function requires the complete knowledge of the space topology and imposes to set limits on

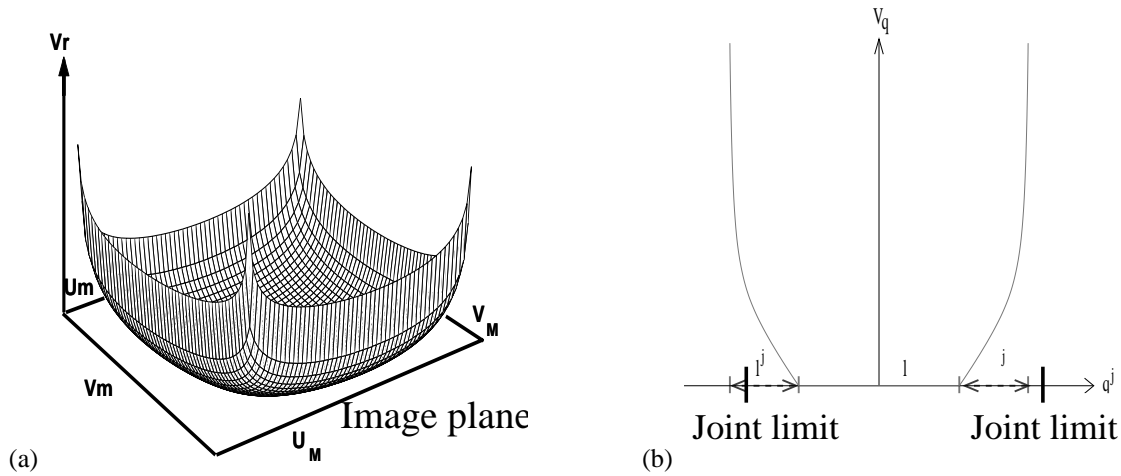


Fig. 17. (a) Repulsive potential for visibility. (b) Repulsive potential for joint limits avoidance.



Fig. 18. (a) Initial and (b) desired images.

the considered workspace. Setting these limits is difficult in practice. Finally, many advantages of the approach proposed in this paper would be lost: robustness with respect to modeling errors, and application to object with unknown CAD model.

8.2. Experimental Results

The proposed method has been tested using the same simple object as before. The images corresponding to the desired and initial camera positions are given in Figures 18(a) and (b), respectively. The corresponding camera displacement is very large ($t_x = 82$ cm, $t_y = 80$ cm, $t_z = 45$ cm, $(u\theta)_x = 37^\circ$, $(u\theta)_y = 45^\circ$, $(u\theta)_z = 125^\circ$). In this case, classical image-based and position-based visual servoing fail.

On all the following plots, joint positions are normalized between $[-1; 1]$, where -1 and 1 represent the joint limits.

In order to emphasize the importance of the introduced constraints in the trajectories, we first perform the path plan-

ning without repulsive potential. The results are given in Figure 19. We can see that the visual features get out largely from the camera field of view (Figure 19(a)) and axis q_5 attains its joint limit (Figure 19(b)). Then, only the repulsive potential associated to the visibility constraint has been activated (see Figure 20). In that case, even if the visibility constraint is ensured (Figure 20(a)), the servoing cannot be realized because axis q_5 reaches its joint limit (Figure 20(b)). In Figure 21, the two repulsive potentials are activated. The target remains in the camera field of view (see Figures 21(a) and (c)) and all axes avoid their joint limit (see Figures 21(b) and (d)). We can notice that the planned trajectories and the realized trajectories in the image are almost similar, which shows the efficiency of our control scheme. The error on the image coordinates of each target point between its current and desired location is given in Figure 21(f). We can note the convergence of the coordinates to their desired value, which demonstrates the correct realization of the positioning task.

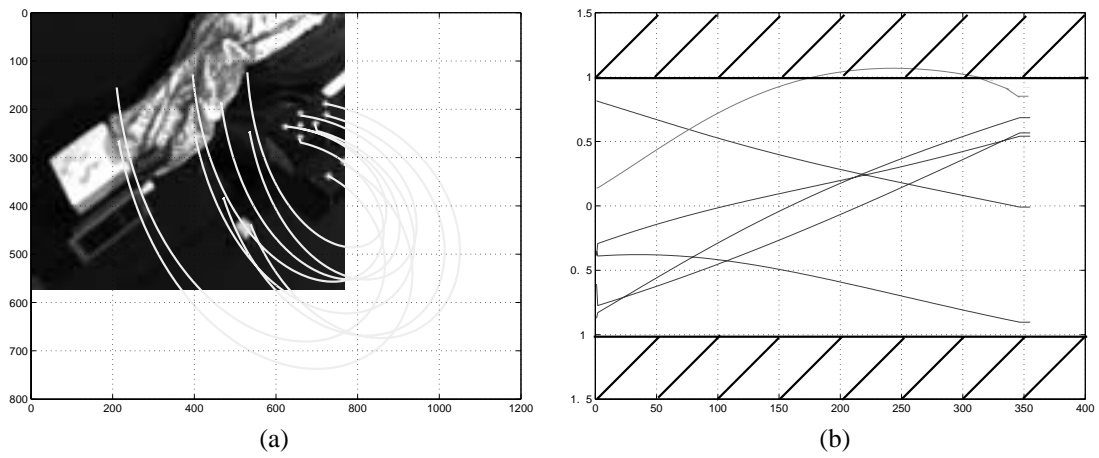


Fig. 19. Planned trajectories without any repulsive potential: (a) in the image; (b) in the joint space.

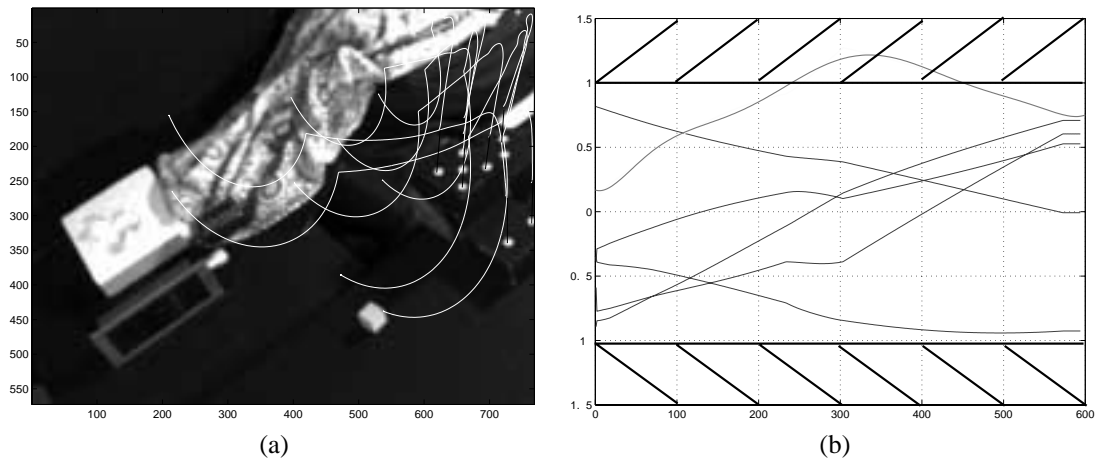


Fig. 20. Planned trajectories without repulsive potential associated to the joint limits avoidance: (a) in the image; (b) in the joint space.

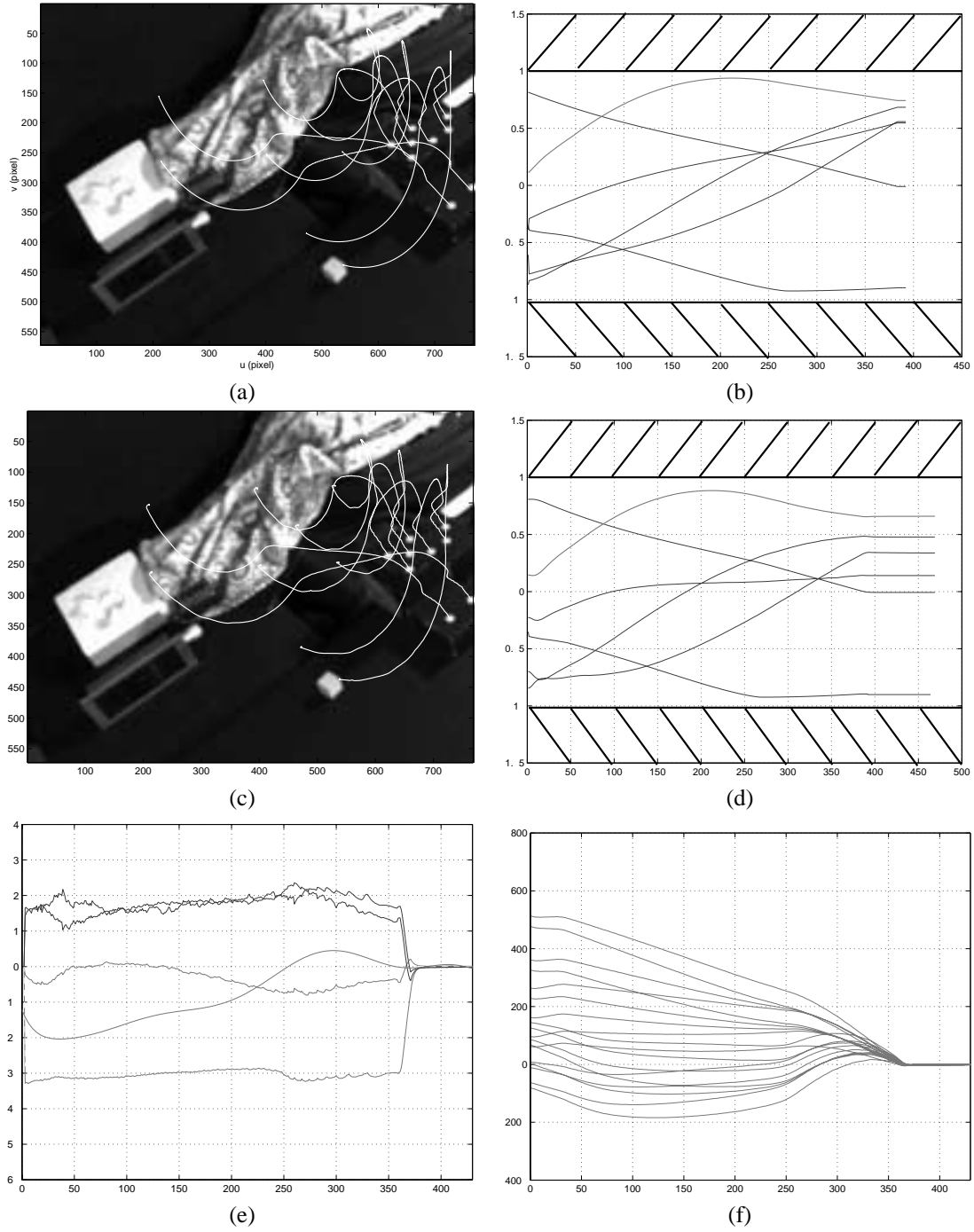


Fig. 21. Planned trajectories with both repulsive potentials: (a) in the image; (b) in the joint space, realized trajectories; (c) in the image; (d) in the joint space; (e) camera translational (cm s^{-1}) and rotational (deg s^{-1}) velocities versus iteration number; (f) errors in the image versus iteration number.

A second experiment is presented in the extensions to this paper and confirms the previous comments (refer to the Appendix for details). The initial and desired images are given by Extensions 4 and 5, respectively. The corresponding robot positions are given by Extensions 6 and 7. The planned trajectory is included in the initial image (Extension 4). As can be seen, the motion to realize is very large and in this case classical approaches fail. Extension 8 is a video acquired by the controlled camera during the positioning task (refer to the Appendix). Extension 9 is a video acquired by an external camera observing the robot motions.

9. Conclusion

In this paper, we have addressed the problem of finding the trajectories in the image space of visual features so that the camera trajectory is optimal. The obtained camera trajectory corresponds to a minimum geodesic in $SE(3)$. The method is model-free and the uncalibrated case has been studied. By coupling the path planning step with an image-based servoing, the proposed method improves significantly the behavior of image-based servoing when the displacement to realize is large. We have validated our approach in a 6-DoF robotic platform by realizing positioning tasks with respect to an unknown target. The experimental results confirm the robustness of our approach with respect to modeling errors. The interpolation of multi-relay images has also been studied in order to cope with the difficult matching problem for a complex scene when the initial image is very different from the desired one. A potential-based path planning has also been described. It allows us to integrate easily constraints in the desired trajectories. A natural perspective for this work is the incorporation of obstacle avoidance in the path planning process as well as the incorporation of non-holonomic constraints.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at <http://www.ijrr.org>.

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Positioning task described in Section 6.2 (correctly calibrated system) seen by the controlled camera. The red and blue crosses represent respectively the current and desired features. The green crosses represent the current desired position of the features (that is their planned trajectories).
2	Video	Same example with badly calibrated camera.

3	Video	Positioning task described in Section 7.2 seen by the controlled camera.
4	Image	Initial image with planned trajectories (Section 8.2).
5	Image	Desired image (Section 8.2).
6	Image	Initial robot configuration (Section 8.2).
7	Image	Final robot configuration (Section 8.2).
8	Video	Motion during the positioning task (Section 8.2) seen by a camera observing the robot.
9	Video	Same example seen by the controlled camera (Section 8.2). The blue and green crosses represent respectively the current and desired features. The red crosses represent the current desired position of the features (i.e., their planned trajectories).

Acknowledgments

The authors wish to thank Anthony Remazeilles and Patrick Gros for their contributions to Section 7. They allowed us to solve the problem of retrieving and delivering images from a large database using image reference query.

References

- Chaumette, F. 1998. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and A. Morse, eds., *The Confluence of Vision and Control*, Lecture Notes in Control and Information Sciences Vol. 237. Springer Verlag, Berlin, pp. 66–78.
- Chesi, G., Hashimoto, K., Prattichizzo, D., and Vicino, A. 2002. Keeping features in the camera's field of view: A visual servoing strategy. *15th International Symposium on Mathematical Theory of Networks and Systems*, Notre-Dame, Indiana.
- Chesi, G., Malis, E., and Cipolla, R. 2000. Automatic segmentation and matching of planar contours for visual servoing. In *IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 630–635.
- Cowan, N. J., and Koditschek, D. E. 1999. Planar image based visual servoing as a navigation problem. In *IEEE International Conference on Robotics and Automation*, Detroit, MI, pp. 611–617.

- Cowan, N. J., Lopes, G. A. D., and Koditschek, D. E. 2000. Rigid body visual servoing using navigation functions. In *IEEE International Conference on Decision and Control*, Sydney, Australia, pp. 3920–3926.
- De Marsico, M., Cinque, L., and Levialdi, S. 1997. Indexing pictorial documents by their content: a survey of current techniques. *Image and Vision Computing* 15(2):119–141.
- Espiau, B., Chaumette, F., and Rives, P. 1992. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation* 8(3):313–326.
- Faugeras, O. 1993. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT press, Cambridge, MA.
- Faugeras, O. and Lustman, F. 1988. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence* 2(3):485–508.
- Harris, C., and Stephens, M. 1988. A combined corner and edge detector. In *Alvey Conference*, pp. 189–192.
- Hashimoto, K. 1993. *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, Singapore.
- Hashimoto, K., and Noritugu, T. 2000. Potential switching control in visual servo. In *IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 2765–2770.
- Hosoda, K., Sakamoto, K., and Asada, M. 1995. Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1(3), pp. 29–34.
- Hutchinson, S., Hager, G. D., and Corke, P. I. 1996. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation* 12(5):651–670.
- Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research* 5(1):90–98.
- Latombe, J. C. 1991. *Robot Motion Planning*. Kluwer Academic, Dordrecht.
- Malis, E. and Chaumette, F. 2000. 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.
- Malis, E., Chaumette, F., and Boudet, S. 1999. 2 1/2 D visual servoing. *IEEE Transactions on Robotics and Automation* 15(2):238–250.
- Mezouar, Y. 2001. *Planification de trajectoires pour l'asservissement visuel*. Ph.D. Thesis, Université de Rennes 1, IRISA.
- Mezouar, Y., and Chaumette, F. 2001. Model-free optimal trajectories in the image space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'01*, Maui, Hawaii, pp. 25–31.
- Mezouar, Y., and Chaumette, F. 2002. Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation* 18(4):534–549.
- Mezouar, Y., Remazeilles, A., Gros, P., and Chaumette, F. 2002. Image interpolation for image-based control under large displacement. In *IEEE International Conference on Robotics and Automation, ICRA '02*, Vol. 3, pp. 3787–3794, Washington, D.C.
- Park, F. C., and Ravani, B. 1997. Smooth invariant interpolation of rotations. *ACM Transactions on Graphics* 16(3):277–295.
- Rimon, E., and Koditschek, D. E. 1992. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation* 8(5):501–518.
- Robert, L., Zeller, C., Faugeras, O. D., and Hebert, M. 1995. Applications of non-metric vision to some visually-guided robotics tasks. Technical Report 2584, INRIA.
- Ruf, A., and Horaud, R. 1997. Visual trajectories from uncalibrated stereo. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'97*, pp. 83–91.
- Samson, C., Espiau, B., and Le Borgne, M. 1991. *Robot Control: The Task Function Approach*. Oxford University Press, Oxford.
- Schmid, C., and Mohr, R. 1997. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(5):530–534.
- Singh, S., Voyle, R. M., Littau, D., and Papanikolopoulos, N. P. 1998. Alignment of an eye-in-hand system to real objects using virtual images. In *IEEE Workshop on Robust Vision for Vision-Based Control of Motion*, Leuven, Belgium.
- Viéville, T., Zeller, C., and Robert, L. 1996. Using collineations to compute motion and structure in an uncalibrated image sequence. *International Journal of Computer Vision* 20(3):213–242.
- Zhang, Z., and Hanson, A. R. 1995. Scaled Euclidean 3D reconstruction based on externally uncalibrated cameras. In *IEEE Symposium on Computer Vision*, Coral Gables, FL.
- Zhang, H., and Ostrowski, J. 2000. Optimal motion planning in the image plane for mobile robots. In *IEEE International Conference on Robotics and Automation*, San Francisco, CA.
- Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.-T. 1995. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal* 78:87–119.