Robotics and Autonomous Systems 41 (2002) 77–87

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Avoiding self-occlusions and preserving visibility by path planning in the image

Youcef Mezouar[*], François Chaumette

IRISA, Campus de Beaulieu, 35042 Rennes, France

Abstract

In this paper, we address the problem of generating trajectories of some image features in order to control efficiently a robotic system using an image-based control strategy. First, physically valid C^2 image trajectories which correspond to quasi-optimal 3D camera trajectory are performed. *Both self-occlusion avoidance and visibility constraints* are taken into account at the task planning level. The good behavior of image-based control when desired and current camera positions are close is then exploited to design an efficient control scheme. Real-time experimental results using a camera mounted on the end effector of a six degree-of-freedom robot confirm the validity of our approach.

Keywords: Visual servoing; Path planning; Visibility; Occlusions

1. Introduction

Visual servoing schemes are traditionally classified into two groups, namely, image-based and position-based [6,8]. Classical approaches are point-to-point-based, that is the robot must reach a desired goal configuration starting from a given initial configuration. In such approaches, a globally stabilizing feedback control solution is required. However if the initial error is large, such a control may product erratic behavior especially in presence of modeling errors. For a very simple case, Cowan and Koditschek [3] describe a globally stabilizing method using navigation function. By composing the error function of 3D Cartesian features and image features, Malis and Chaumette [11] propose a globally stabilizing solution called $2\frac{1}{2}$D visual servoing for general setup. Classical image-based visual servoing is a local control solution. It thus requires the definition of intermediate subgoals in the sensor space at the task planning level if the initial error is large. In this approach, the robot effector is controlled so that the image features converge to the reference image features. The robot effector trajectory in the 3D Cartesian space is not controlled. Such a control can thus provide inadequate robot motion leading to no optimal or no physically valid robot trajectory [1]. However, it is well known that image-based control is locally stable and robust with respect to modeling errors and noise perturbations. The key idea of our work is to use the local stability and robustness of image-based servoing by specifying trajectories to follow in the image. Indeed, for a trajectory following a local control solution works properly since current and desired configurations remain close. Only few papers deal with path planning in image space. In [7], a trajectory generator

[*] Corresponding author.
E-mail addresses: ymezouar@cs.columbia.edu (Y. Mezouar), francois.chaumette@irisa.fr (F. Chaumette).

0921-8890/02/$ – see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00277-4

using a stereo system is proposed and applied to obstacle avoidance. An alignment task using intermediate views of the object synthesized by image morphing is presented in [18]. A path planning for a straight-line robot translation observed by a weakly calibrated stereo system is performed in [16]. In previous work [13], we have proposed a potential field-based path planning generator that determines the trajectories in the image of a set of points lying on a planar target. In [14], this methodology was extended to non-planar objects and applied to the joint limit avoidance. In this paper, we propose to plan the trajectory of an unknown and not necessarily planar object. Both self-occlusions and visibility constraints are taken into account. Contrary to others approaches [2,15] exploiting the robot redundancy, the self-occlusions and visibility constraints can be ensured even if all the robot degrees of freedom are used to realize the task.

More precisely, we plan the trajectory of $\mathbf{s} = [\mathbf{p}_1^{\mathrm{T}}, \dots, \mathbf{p}_n^{\mathrm{T}}]^{\mathrm{T}}$, composed of the $2 \times n$ image coordinates of n points $\mathcal{M}_j$ lying on an unknown target, between the initial configuration $\mathbf{s}_i = [\mathbf{p}_{1i}^{\mathrm{T}}, \dots, \mathbf{p}_{ni}^{\mathrm{T}}]^{\mathrm{T}}$ and the desired one $\mathbf{s}^* = [\mathbf{p}_1^{*\mathrm{T}}, \dots, \mathbf{p}_n^{*\mathrm{T}}]^{\mathrm{T}}$. Our approach consists of three phases. In the first one, the discrete geometric camera path (that ensures the physical validity of the image trajectories) is performed as a sequence of N intermediate camera poses which approaches as much as possible a straight line in the Cartesian space. In this phase, the self-occlusion avoidance and the visibility constraint are introduced. In the second one, the discrete geometric trajectory of the target in the image and the discrete geometric trajectory of the robot in the joint space are obtained from the camera path. Finally, continuous and derivable geometric paths in the image with an associated timing law $\mathbf{s}^*(t)$ are generated and tracked using an image-based control scheme.

The paper is organized as follows. In Section 2, we propose a modification of the potential function method to integrate constraints defined in different spaces. The method of path planning is presented in Section 3. In Section 4, we show how to use an image-based control approach to track the trajectories. In Section 5, a timing law is associated with the obtained geometric path. The experimental results are given in Section 6.

2. Modified potential field method

Our path planning strategy is based on the potential field method. This method was originally developed for an on-line collision avoidance [9,10].

2.1. Classical approach

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 toward the goal configuration ($\boldsymbol{\Upsilon}_*$) 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 $\mathbf{F}(\boldsymbol{\Upsilon})$, where the 6×1 vector $\boldsymbol{\Upsilon}$ represents a parameterization of the robot workspace $\mathcal{W} \subset \mathbb{R}^p$, is induced by the potential function. This force is defined as $\mathbf{F}(\boldsymbol{\Upsilon}) = -\vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V$, where $\vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V$ denotes the transpose of the gradient vector of V at $\boldsymbol{\Upsilon}$. Using these conventions, $\mathbf{F}(\boldsymbol{\Upsilon})$ can be decomposed as the sum of two vectors, $\mathbf{F}_{\mathrm{a}}(\boldsymbol{\Upsilon}) = -\vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V_{\mathrm{a}}$ and $\mathbf{F}_{\mathrm{r}}(\boldsymbol{\Upsilon}) = -\vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V_{\mathrm{r}}$, which are respectively called the attractive and repulsive forces. Path generation proceeds along the direction of $\mathbf{F}(\boldsymbol{\Upsilon})$ 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:

$$\boldsymbol{\Upsilon}_{k+1} = \boldsymbol{\Upsilon}_k + \varepsilon_k \frac{\mathbf{F}(\boldsymbol{\Upsilon}_k)}{\|\mathbf{F}(\boldsymbol{\Upsilon}_k)\|}, \tag{1}$$

where k is the increment index and ε_k is a positive scaling factor denoting the length of the kth increment. To deal with constraints defined in various spaces and to control more efficiently the trajectories, we use a modified potential field method.

2.2. Modified forces

Consider the unconstrained problem:

$$\min V(\boldsymbol{\Upsilon}), \quad \boldsymbol{\Upsilon} \in \mathbb{R}^p.$$

A classical continuous gradient strategy for finding a minimum of V consists of making $\boldsymbol{\Upsilon}(t)$ vary according to the evolution equation:

$$\dot{\boldsymbol{\Upsilon}} = -\epsilon \mathbf{Q} \vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V, \tag{2}$$

where ϵ is a positive scalar and $\mathbf{Q}$ is a constant positive matrix. Premultiplying (2) by $\vec{\nabla}_{\boldsymbol{\Upsilon}} V$, we get:

$$\frac{\mathrm{d}}{\mathrm{d}t} V(\boldsymbol{\Upsilon}) = -\epsilon \vec{\nabla}_{\boldsymbol{\Upsilon}} V \mathbf{Q} \vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V \leq 0. \tag{3}$$

Thus V decreases with time as long as $\vec{\nabla}_{\boldsymbol{\Upsilon}} V \neq 0$, and remains constant when $\vec{\nabla}_{\boldsymbol{\Upsilon}} V = 0$. A common and simple choice for $\mathbf{Q}$ is the identity matrix $\mathbf{I}$. In this case, $\boldsymbol{\Upsilon}$ moves in the direction opposite to the gradient at $\boldsymbol{\Upsilon}$. This strategy is adopted in the classical approach described previously, where $\mathbf{F} = -\vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V$. Consider now a potential field $V_{\mathbf{f}} = V(\mathbf{f}(\boldsymbol{\Upsilon}))$, where $\mathbf{f}$ is differentiable everywhere in $\mathcal{W}$. The evolution equation of $\mathbf{f}$, when $\boldsymbol{\Upsilon}$ moves according to (2), is given by

$$\begin{aligned}\dot{\mathbf{f}} &= -\epsilon \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\Upsilon}}\right) \mathbf{Q} \vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V \\ &= -\epsilon \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\Upsilon}}\right) \mathbf{Q} \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\Upsilon}}\right)^{\mathrm{T}} \vec{\nabla}_{\mathbf{f}}^{\mathrm{T}} V. \end{aligned} \tag{4}$$

In order that $\mathbf{f}$ moves in the direction opposite to the gradient of V at $\mathbf{f}$, the matrix $\mathbf{Q}$ can be chosen adequately:

$$\mathbf{Q} = \mathbf{Q}_{\mathbf{f}} = \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\Upsilon}}\right)^{+} \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\Upsilon}}\right)^{+\mathrm{T}}. \tag{5}$$

Note that $\mathbf{Q}$ is a positive matrix and thus the relation (3) is verified. The evolution equation (4) can thus be rewritten as:

$$\dot{\mathbf{f}} = -\epsilon \vec{\nabla}_{\mathbf{f}}^{\mathrm{T}} V.$$

The artificial force associated to the potential field $V_{\mathbf{f}}(\mathbf{f}(\boldsymbol{\Upsilon}))$ is thus:

$$\mathbf{F}_{\mathbf{f}}(\boldsymbol{\Upsilon}) = -\mathbf{Q} \vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V_{\mathbf{f}} = -\left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\Upsilon}}\right)^{+} \vec{\nabla}_{\mathbf{f}}^{\mathrm{T}} V_{\mathbf{f}}. \tag{6}$$

When several potential functions are considered, the dominant artificial force derived from the potential $V_{\mathbf{f}}$ creates a dominant motion of $\mathbf{f}$ in the direction opposite to the gradient of $V_{\mathbf{f}}$ at $\mathbf{f}$. In practice, by using such process, it is more easy to control the relative influence of each force and thus to control the camera or the object trajectories. 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 self-occlusions of the target image features are not allowed.

To deal with the first constraint, the motion is firstly planned in the 3D Cartesian space and then projected in the image space. The attractive potential ($V_{\boldsymbol{\Upsilon}}$) pulling the robot toward the goal configuration ($\boldsymbol{\Upsilon}_*$) is thus defined in the 3D Cartesian space. The second and the third constraints are introduced through two repulsive potentials $V_{1\mathbf{s}}$ and $V_{2\mathbf{s}}$ defined in the image. The total force is given by

$$\mathbf{F} = \mathbf{F}_{\boldsymbol{\Upsilon}} + \gamma \mathbf{F}_{1\mathbf{s}} + \chi \mathbf{F}_{2\mathbf{s}}, \tag{7}$$

where the scaling factors γ and χ allow us to adjust the relative influence of the different forces and can thus be used to take out of potential local minima. If a local minimum is reached, a motion is executed to take out of it by favoring the repulsive force. According to (6), the artificial forces can be written as follows:

$$\begin{aligned} \mathbf{F}_{\boldsymbol{\Upsilon}} &= -\left(\frac{\partial \boldsymbol{\Upsilon}}{\partial \boldsymbol{\Upsilon}}\right)^{+} \vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V_{\boldsymbol{\Upsilon}} = -\vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V_{\boldsymbol{\Upsilon}}, \\ \mathbf{F}_{1\mathbf{s}} &= -\left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\Upsilon}}\right)^{+} \vec{\nabla}_{\mathbf{s}}^{\mathrm{T}} V_{1\mathbf{s}} = -\mathbf{M}^{+}\mathbf{L}^{+} \vec{\nabla}_{\mathbf{s}}^{\mathrm{T}} V_{1\mathbf{s}}, \\ \mathbf{F}_{2\mathbf{s}} &= -\left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\Upsilon}}\right)^{+} \vec{\nabla}_{\mathbf{s}}^{\mathrm{T}} V_{2\mathbf{s}} = -\mathbf{M}^{+}\mathbf{L}^{+} \vec{\nabla}_{\mathbf{s}}^{\mathrm{T}} V_{2\mathbf{s}}, \end{aligned} \tag{8}$$

where $\mathbf{M}$ is the Jacobian matrix that relates the variation of the camera velocity $\mathbf{T}_{\mathrm{c}} = \dot{\mathbf{r}}$ to the variation of the chosen parameterization $\boldsymbol{\Upsilon}$: $\mathbf{T}_{\mathrm{c}} = \mathbf{M}\dot{\boldsymbol{\Upsilon}}$. The form of the matrix $\mathbf{M}$ will be given in the sequel for the chosen workspace parameterizations. 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 $\mathbf{T}_{\mathrm{c}}$: $\dot{\mathbf{s}} = \mathbf{L}\mathbf{T}_{\mathrm{c}}$. For a point $\mathcal{M}^j$ with coordinates $[X^j, Y^j, Z^j]^{\mathrm{T}}$ in the current camera frame and coordinates $\mathbf{p}^j = [u^j, v^j, 1]^{\mathrm{T}}$, the interaction matrix $\mathbf{L}(\mathbf{p}^j, Z^j)$ related to $\mathbf{s} = [x_j, y_j]^{\mathrm{T}}$ is given by

$$\mathbf{a}\begin{bmatrix} -\dfrac{1}{Z^j} & 0 & \dfrac{x^j}{Z^j} & x^j y^j & -(1+x^{j^2}) & y^j \\ 0 & -\dfrac{1}{Z^j} & \dfrac{y^j}{Z^j} & (1+y^{j^2}) & -x^j y^j & -x^j \end{bmatrix},$$

where $[x_j, y_j, 1]^{\mathrm{T}} = \mathbf{A}^{-1}\mathbf{p}^j$. The matrix $\mathbf{A}$ denotes a non-singular matrix containing the camera internal parameters, the matrices $\mathbf{A}$ and $\mathbf{a}$ are given by

$$\mathbf{A} = \begin{bmatrix} fp_u & -fp_u\cot(\theta) & u_0 \\ 0 & \dfrac{fp_v}{\sin(\theta)} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \multicolumn{2}{c}{\mathbf{a}} & \begin{matrix} u_0 \\ v_0 \end{matrix} \\ 0 & 0 & 1 \end{bmatrix},$$

where u_0 and v_0 are the pixels coordinates of principal point, f the focal length, p_u and p_v the magnifications, respectively, in the u- and v-directions and θ is the angle between these axes. When $\mathbf{s}$ is composed of the image coordinates of n points, the corresponding interaction matrix is

$$\mathbf{L}(\mathbf{s}, \mathbf{Z}) = [\mathbf{L}^{\mathrm{T}}(\mathbf{p}^1, Z^1), \ldots, \mathbf{L}^{\mathrm{T}}(\mathbf{p}^n, Z^n)]^{\mathrm{T}}.$$

3. Trajectory planning

We consider that the target model is not available. In this case, the camera pose cannot be estimated. Only a scaled Euclidean reconstruction can be obtained by performing a partial pose estimation as described in the next subsection. This partial pose estimation and the relations linking two views of a static object are then exploited to design a path of the projection of the unknown object in the image.

3.1. Scaled Euclidean reconstruction

Let $\mathcal{F}^*$ and $\mathcal{F}$ be the frames attached to the camera in its desired and current positions. The rotation matrix and the translation vector between $\mathcal{F}$ and $\mathcal{F}^*$ are denoted ${}^*\mathbf{R}_{\mathrm{c}}$ and ${}^*\mathbf{t}_{\mathrm{c}}$, respectively. A target point $\mathcal{M}_j$ with homogeneous coordinates $\mathbf{M}_j = [X_j, Y_j, Z_j, 1]$ (resp. $\mathbf{M}_j^*$) in $\mathcal{F}$ (resp. $\mathcal{F}^*$) is projected in the camera image onto a point with homogeneous normalized and pixel coordinates $\mathbf{m}_j = [x_j, y_j, 1]^{\mathrm{T}}$ (resp. $\mathbf{m}_j^*$) and $\mathbf{p}_j = [u_j, v_j, 1]^{\mathrm{T}} = \mathbf{A}m_j$ (resp. $\mathbf{p}_j^*$). Consider a 3D reference plane $\mathit{\Pi}$ given in $\mathcal{F}^*$ by $\boldsymbol{\pi}^{\mathrm{T}} = [\mathbf{n}^* \; -d^*]$, where $\mathbf{n}^*$ is its unitary normal in $\mathcal{F}^*$ and d^* is the distance from $\mathit{\Pi}$ to the origin of $\mathcal{F}^*$ (see Fig. 1). It is well known that there is a projective homography matrix $\mathbf{G}$ such that:

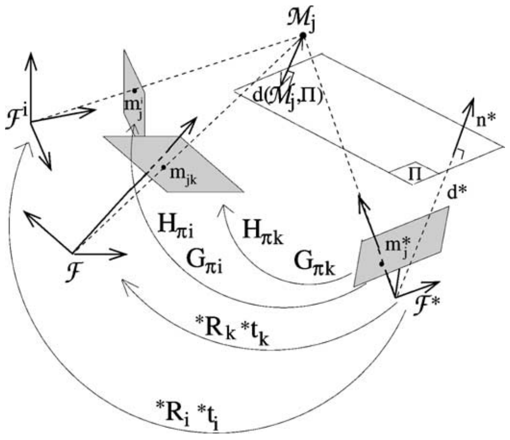


Fig. 1. Scaled camera trajectory.

$$\alpha_j \mathbf{p}_j = \mathbf{G}\mathbf{p}_j^* + \beta_j \mathbf{e} \quad \text{with } \mathbf{e} = -\mathbf{A}\,{}^*\mathbf{R}_{\mathrm{c}}^{\mathrm{T}}\,{}^*\mathbf{t}_{\mathrm{c}}, \tag{9}$$

where α_j is a positive scaling factor and β_j is a scaling factor null if the target point is linked with $\mathit{\Pi}$. More precisely, if we define the signed distance $d_j = d(\mathcal{M}_j, \mathit{\Pi}) = \boldsymbol{\pi}\mathbf{M}_j^*$, we have:

$$\beta_j = -\frac{d_j}{Z_j^* d^*}. \tag{10}$$

If at least four matched points belonging to $\mathit{\Pi}$ are known, $\mathbf{G}_\pi$ can be estimated by solving a linear system. Else, at least eight points (three points to define $\mathit{\Pi}$ and five outside of $\mathit{\Pi}$) are necessary to estimate the homography matrix by using for example the linearized algorithm proposed in [11]. Assuming that the camera calibration is known, the Euclidean homography $\mathbf{H}$ of plane $\mathit{\Pi}$ is estimated as follows:

$$\mathbf{H} = \mathbf{A}^{-1}\mathbf{G}\mathbf{A} \tag{11}$$

and it can be decomposed into a rotation matrix and a rank 1 matrix [5]:

$$\mathbf{H} = {}^*\mathbf{R}_{\mathrm{c}}^{\mathrm{T}} - {}^*\mathbf{R}_{\mathrm{c}}^{\mathrm{T}}\mathbf{t}_{d^*}\mathbf{n}^{*\mathrm{T}} \quad \text{with } \mathbf{t}_{d^*} = \frac{{}^*\mathbf{t}_{\mathrm{c}}}{d^*}. \tag{12}$$

From $\mathbf{H}$ and the image features, it is thus possible to determine the camera motion parameters (that is the rotation ${}^*\mathbf{R}_{\mathrm{c}}$ and the scaled translation $\mathbf{t}_{d^*}$) and the vector $\mathbf{n}^*$ [5]. The structure of the observed scene can also be determined. For example, the ratio between the Z-coordinates of a 3D point $\mathcal{M}_j$ expressed in $\mathcal{F}$ and the distance d^*, $\rho_j = Z_j/d^*$ can be obtained from ${}^*\mathbf{R}_{\mathrm{c}}$, $\mathbf{t}_{d^*}$ and the image features [11].These parameters are important since they are used in the path planning generator and in the control scheme.

3.2. *Scaled 3D Cartesian trajectory*

The initial homography matrix $\mathbf{G}_i$ is computed from $\mathbf{s}^{\mathrm{i}}$ and $\mathbf{s}^*$. According to (11), we obtain $\mathbf{H}_i$. Then $\mathbf{n}^*$ is estimated, as well as the rotation ${}^*\mathbf{R}_i$ and the scaled translation $\mathbf{t}_{d^*i} = {}^*\mathbf{t}_i/d^*$ between $\mathcal{F}^i$ (frame linked to the camera in its initial position) and $\mathcal{F}^*$. If we choose as partial parameterization of the workspace $\boldsymbol{\Upsilon} = [\mathbf{t}_{d^*}^{\mathrm{T}}, (\mathbf{u}\theta)^{\mathrm{T}}]^{\mathrm{T}}$, where $\mathbf{u}$ and θ are the normalized rotation axis and the rotation angle extracted from ${}^*\mathbf{R}_{\mathrm{c}}$, we obtain at the initial and desired robot configurations $\boldsymbol{\Upsilon}_i = [\mathbf{t}_{d^*i}^{\mathrm{T}}, (\mathbf{u}\theta)_i^{\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{\Upsilon}_* = \mathbf{0}_{6\times 1}$. The camera path starting at the initial configuration $\boldsymbol{\Upsilon}_{k=0} = \boldsymbol{\Upsilon}_i$ and ending at $\boldsymbol{\Upsilon}_* = \mathbf{0}_{6\times 1}$ is obtained using the transition equation (1) where the artificial force will be defined in the sequel.

3.3. *Image trajectories*

The homography matrix $\mathbf{G}_k$ of plane Π relating the current and desired images can be computed from $\boldsymbol{\Upsilon}_k$ using (11) and (12):

$$\mathbf{G}_k = \mathbf{A}({}^*\mathbf{R}_k^{\mathrm{T}} - {}^*\mathbf{R}_k^{\mathrm{T}}\mathbf{t}_{d^*k}\mathbf{n}^{*\mathrm{T}})\mathbf{A}^{-1}. \tag{13}$$

According to (9), the image coordinates of the points $\mathcal{M}_j$ at time k are given by

$$\mu_{jk}\mathbf{p}_{jk} = \mathbf{G}_k\mathbf{p}_j^* + \beta_j\mathbf{e}_k, \tag{14}$$

where (refer to (9) and (10))

$$\beta_j\mathbf{e}_k = \frac{d_j}{Z_j^*}\mathbf{A}^*\mathbf{R}_k^{\mathrm{T}}\mathbf{t}_{d^*k}.$$

Using the previous relation, (14) can be rewritten as:

$$\mu_{jk}\mathbf{p}_{jk} = \mathbf{G}_k\mathbf{p}_j^* + \frac{d(\mathcal{M}_j, \Pi)}{Z_j^*}\mathbf{A}^*\mathbf{R}_k^{\mathrm{T}}\mathbf{t}_{d^*k}. \tag{15}$$

Furthermore, if the relation (15) is applied between the desired and initial camera positions, we obtain:[1]

$$\frac{d_j}{Z_j^*} = \mathrm{sign}\left(\frac{(\mu_j\mathbf{p}_{ji} - \mathbf{G}_i\mathbf{p}_j^*)_1}{(\mathbf{A}^*\mathbf{R}_i^{\mathrm{T}}\mathbf{t}_{d^*i})_1}\right)\frac{\|\mathbf{G}_i\mathbf{p}_j^* \wedge \mathbf{p}_{ji}\|}{\|\mathbf{A}^*\mathbf{R}_i^{\mathrm{T}}\mathbf{t}_{d^*i} \wedge \mathbf{p}_{ji}\|}. \tag{16}$$

Eqs. (13), (15) and (16) allow to compute $\mu_{jk}\mathbf{p}_{jk}$ from $\boldsymbol{\Upsilon}_k$ and the initial and desired visual features. The image coordinates $\mathbf{p}_{jk}$ are then computed by dividing $\mu_j\mathbf{p}_{jk}$ by its last component. Furthermore, the ratio ρ_{jk}, which will be used in the repulsive force and in the control law, can easily be obtained from $\boldsymbol{\Upsilon}_k$ and $\mathbf{m}_{jk} = \mathbf{A}^{-1}\mathbf{p}_{jk}$.

3.4. *Reaching the goal*

The attractive potential field $V_{\boldsymbol{\Upsilon}}$ is simply defined as a parabolic function in order to minimize the distance between the current position and the desired one:

$$V_{\boldsymbol{\Upsilon}}(\boldsymbol{\Upsilon}) = \tfrac{1}{2}\|\boldsymbol{\Upsilon} - \boldsymbol{\Upsilon}_*\|^2 = \tfrac{1}{2}\|\boldsymbol{\Upsilon}\|^2.$$

The function $V_{\boldsymbol{\Upsilon}}$ is positive or null and attains its minimum at $\boldsymbol{\Upsilon}_*$, where $V_{\boldsymbol{\Upsilon}}(\boldsymbol{\Upsilon}_*) = 0$. It generates a force $\mathbf{F}_{\boldsymbol{\Upsilon}}$ that converges linearly toward the goal configuration:

$$\mathbf{F}_{\boldsymbol{\Upsilon}}(\boldsymbol{\Upsilon}) = -\vec{\nabla}_{\boldsymbol{\Upsilon}}^{\mathrm{T}} V_{\boldsymbol{\Upsilon}} = -\boldsymbol{\Upsilon}. \tag{17}$$

When the repulsive potentials are not needed, the transition equation can be written as (refer to (1) and (17)):

$$\boldsymbol{\Upsilon}_{k+1} = \left(1 - \frac{\varepsilon_k}{\|\boldsymbol{\Upsilon}_k\|}\right)\boldsymbol{\Upsilon}_k.$$

Thus, $\boldsymbol{\Upsilon}_k$ is lying on the straight line passing by $\boldsymbol{\Upsilon}_i$ and $\boldsymbol{\Upsilon}_*$. As a consequence, the translation performed by the camera is a real straight line since $\boldsymbol{\Upsilon}_k$ is defined with respect to a motionless frame (that is $\mathcal{F}_*$). However, the object can get out of the camera field of view and occlusions can occur. To avoid this potential problems, two repulsive forces are introduced by deviating the camera trajectory when needed.

[1] $(\mathbf{v})_j$ is the jth components of $\mathbf{v}$.

3.5. Self-occlusions and visibility

3.5.1. Visibility constraint

A point $\mathcal{M}^j$, which projects onto the image plane at a point with image coordinates $\mathbf{p}^j = [u^j, v^j, 1]^{\mathrm{T}}$, is known as visible if $u_j \in [u_m, u_M]$ and $v_j \in [v_m, v_M]$, where u_m, u_M, v_m, v_M are the limits of the image. The vector of image features $\mathbf{s}$ is called acceptable if for all $j \in \{1, \dots, n\}$, $u^j \in [u_m + \alpha; u_M - \alpha]$ and $v^j \in [v_m + \alpha; v_M - \alpha]$, where α is a positive constant denoting the distance of influence of the image boundary (see Fig. 2(a)). We denote $\mathcal{C}$ the set of acceptable image features. One way to create a potential barrier around the camera field of view, ensuring that all features are always visible and do not affect the camera motion when they are sufficiently far enough from the image limits, is to define the repulsive potential $V_{1\mathbf{s}}(\mathbf{s})$ as a null scalar if $\mathbf{s} \in \mathcal{C}$ and as follow if $\mathbf{s} \notin \mathcal{C}$ (see Fig. 2(b)):

$$-v_{1s}^2 \log \prod_{j=1}^{n} \left(1-\frac{u^j}{u_M}\right)\left(1-\frac{u^j}{u_m}\right)\left(1-\frac{v^j}{v_M}\right) \times \left(1-\frac{v^j}{v_m}\right). \qquad (18)$$

In order to obtain a continuous and derivable potential field $V_{1\mathbf{s}}$, v_{1s} is chosen as a bounded function with null value in the boundary of $\mathcal{C}$:

$$v_{1s}(\mathbf{s}) = \prod_{j=1}^{n} (u^j - u_M^\alpha)(u^j - u_m^\alpha)(v^j - v_M^\alpha)(v^j - v_m^\alpha),$$

where $u_m^\alpha = u_m + \alpha$, $u_M^\alpha = u_M - \alpha$, $v_m^\alpha = v_m + \alpha$ and $v_M^\alpha = v_M - \alpha$. The function $V_{1\mathbf{s}}$ is positive or null, tends to infinity when at less one selected image features gets closer to the image limits, and it is null when all image features are sufficiently far away from the image limits. The artificial repulsive force deriving from $V_{1\mathbf{s}}$ is

$$\mathbf{F}_{1\mathbf{s}}(\boldsymbol{\Upsilon}) = -\mathbf{M}^{+}\mathbf{L}^{+}\vec{\boldsymbol{\nabla}}_{\mathbf{s}}^{\mathrm{T}} V_{1\mathbf{s}}. \qquad (19)$$

$\mathbf{M}$ is the 6×6 Jacobian matrix that relates the variation of the camera velocity $\mathbf{T}_{\mathrm{c}}$ to the variation of $\boldsymbol{\Upsilon}$:

$$\mathbf{M}(d^*) = \begin{bmatrix} d^* \, {}^*\mathbf{R}_k^{\mathrm{T}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{L}_{wk}^{+} \end{bmatrix}.$$

The computation of $\mathbf{L}_{wk}^{+}$ can be found in [12]:

$$\mathbf{L}_{wk}^{+} = \mathbf{I} + \frac{\theta_k}{2}\,\mathrm{sinc}^2\left(\frac{\theta_k}{2}\right)[\mathbf{u}_k]_\wedge + (1 - \mathrm{sinc}(\theta_k))[\mathbf{u}_k]_\wedge^2,$$

where $[\mathbf{u}]_\wedge$ denotes the skew symmetric matrix associated to the vector $\mathbf{u}$. The interaction matrix depends of the depth vector $\mathbf{Z}$. It cannot be computed directly from the scaled parameterization $\boldsymbol{\Upsilon}$. But the ratio $\rho_k^j = Z_k^j / d^*$ can easily be estimated from the scaled parameterization and the image features according to (13). Thus we rewrite the interaction matrix $\mathbf{L}(\mathbf{s}, \mathbf{Z})$ as follows:

$$\mathbf{L}(\mathbf{s}, \boldsymbol{\Gamma}, d^*) = \left[\frac{1}{d^*}\mathbf{S}, \mathbf{Q}\right], \qquad (20)$$

where $\boldsymbol{\Gamma} = [\rho_k^1, \dots, \rho_k^n]$, $\mathbf{S} = [\mathbf{S}^{1\mathrm{T}}, \dots, \mathbf{S}^{n\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{Q} = [\mathbf{Q}^{1\mathrm{T}}, \dots, \mathbf{Q}^{n\mathrm{T}}]^{\mathrm{T}}$ are two $2n \times 3$ matrices independent of d^*:

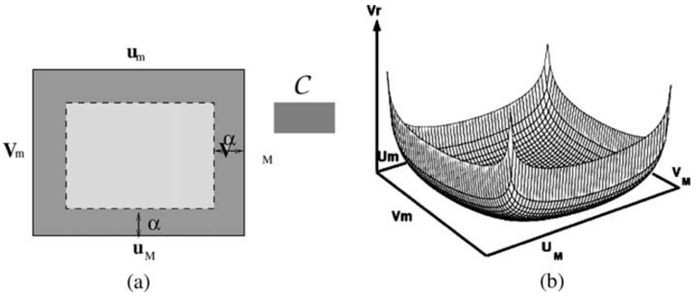


Fig. 2. (a) Image limits, (b) repulsive potential for visibility.

$$\mathbf{S}^j = \mathbf{a}\begin{bmatrix} -\dfrac{1}{\rho_k^j} & 0 & \dfrac{x_k^j}{\rho_k^j} \\ 0 & -\dfrac{1}{\rho_k^j} & \dfrac{y_k^j}{\rho_k^j} \end{bmatrix},$$

$$\mathbf{Q}^j = \mathbf{a}\begin{bmatrix} x^{j_k} y^{j_k} & -1 - x^{j_k^2} & y^{j_k} \\ 1 + y^{j_k^2} & -x^{j_k} y_k^j & -x_k^j \end{bmatrix}.$$

The vector $\vec{\nabla}_{\mathbf{s}}^{\mathrm{T}} V_{1\mathbf{s}}$ is easily obtained from (22).

Remark. Note that the product of the matrices $\mathbf{M}^+$ and $\mathbf{L}^+$ is independent on the unknown parameter d^*. Thus, the artificial force $\mathbf{F}_{1s}$ is also independent of d^*.

3.5.2. Self-occlusion avoidance

The goal is to avoid the occlusions of the target $\mathcal{T}$ by static objects when the camera moves. Let us consider the projection in the image of N parts of the static scene $\mathcal{P}_j$ ($j \in \{1, \ldots, N\}$), we say that an occlusion occurs if:

$$\|\mathcal{P}_j - \mathcal{P}_i\| \leq \zeta, \tag{21}$$

where $i \neq j$ and ζ is a chosen scaling factor. An important class of occlusion are characterized by $\mathcal{P}_j \subset \mathcal{T}$ and $\mathcal{P}_i \subset \mathcal{T}$ and are called self-occlusions (see Fig. 3). The self-occlusions are detected by using Eq. (21). We denote by $\mathcal{O}$ the set of image feature configurations such that:

$$\|\mathcal{P}_j - \mathcal{P}_i\| \leq \zeta + l,$$

where l is a scaling factor. In order to avoid the occlusions, the repulsive potential $V_{2\mathbf{s}}$ is defined as a null function if $\mathbf{s} \in \mathcal{O}$ and as follows if $\mathbf{s} \notin \mathcal{O}$ (see Fig. 3(c)):

$$V_{2\mathbf{s}} = -v_{2s}^2 \log \prod_{\substack{i,j=1 \\ j \neq i}}^{n} \left(1 - \frac{\zeta}{\|\mathcal{P}_j - \mathcal{P}_i\|}\right). \tag{22}$$

As for the previous potential function, v_{2s} is chosen as a bounded function with null value in the boundary of $\mathcal{O}$. The function $V_{2\mathbf{s}}$ is positive or null, tends to infinity when $\mathcal{P}_j$ is near $\mathcal{P}_i$, and it is null if the set of considered image features belongs to $\mathcal{O}$. The associated artificial force is directly obtained from (22) and (8).

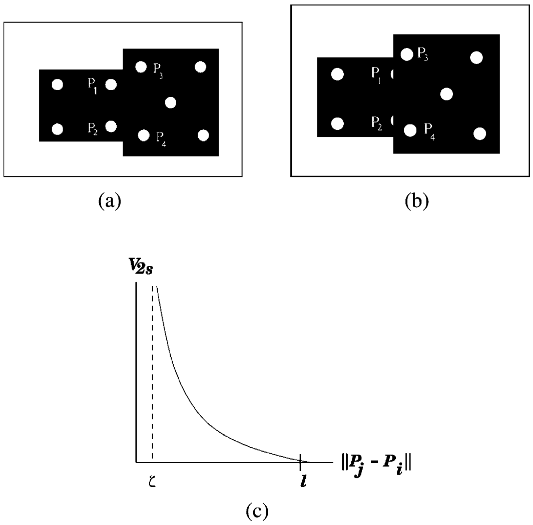


Fig. 3. Image of a rigid target: (a) all the image features are visible, (b) after camera moved part of the image features are occluded, and (c) repulsive potential for self-occlusions avoidance.

4. Performing C^2 timing law

In the previous subsection, we have obtained discrete trajectories. In order to design continuous and derivable curves and thus to improve the dynamic behavior of the system, we use cubic B-spline interpolation. The spline interpolation problem is usually stated as: given data points $\mathcal{S} = \{\mathbf{s}_k | k \in 1, \ldots, N\}$ and a set of parameter values $\mathcal{T} = \{t_k | k \in 1, \ldots, N\}$, we have to determine a cubic B-spline curve $\mathbf{s}(t)$ such that $\mathbf{s}(t_k) = \mathbf{s}_k, \forall t_k$. In practice, parameter values are rarely given. In our case, we can adjust them to the distribution of the vector of image features $\mathbf{s}_k$ or using the distribution of the camera positions $\boldsymbol{\Upsilon}_k$. In order to control efficiently the camera velocity, it is more reasonable to use the distribution of the camera positions. The time values are thus chosen spacing proportionally to the distances between camera positions (see Fig. 4):

$$\frac{\Delta t_k}{\Delta t_{k+1}} = \frac{t_{k+1} - t_k}{t_{k+2} - t_{k+1}} = \frac{\|\boldsymbol{\Upsilon}_{k+1} - \boldsymbol{\Upsilon}_k\|}{\|\boldsymbol{\Upsilon}_{k+2} - \boldsymbol{\Upsilon}_{k+1}\|}.$$

Considering the transition equation (1), we obtain:

$$\Delta t_{k+1} = \frac{\varepsilon_k}{\varepsilon_{k+1}} \Delta t_k \quad \text{with} \quad \Delta t_0 = T.$$

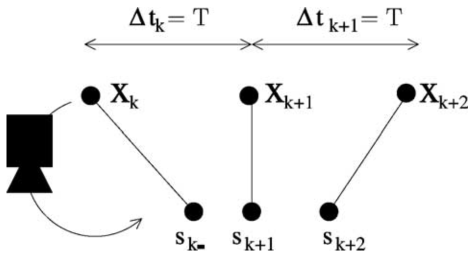


Fig. 4. Controlling the time along the camera trajectory.

T being the time between two consecutive frames (chosen for example as the video rate). In practice, ε_k is chosen constant, we thus have $t_k = kT$. Given the data vectors $\mathbf{s}_k$ and the parameters values t_k, the image data can be interpolated by using a natural cubic B-spline interpolation and we obtain a C^2 function $\mathbf{s}(t)$ defined for $(k-1)\Delta T \leq t \leq k\Delta T$ by

$$\mathbf{s}(t) = \mathbf{A}_k t^3 + \mathbf{B}_k t^2 + \mathbf{C}_k t + \mathbf{D}_k, \tag{23}$$

where the $n \times n$ diagonal matrices $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$ are obtained from $\mathcal{S}$ and $\mathcal{T}$. Finally, the ratio ρ appears in the control law. By using the same process, $\mathbf{\Gamma}(t) = [\rho_1(t), \ldots, \rho_n(t)]$ is computed from $\mathcal{R} = \{\mathbf{\Gamma}_k | k \in 1, \ldots, N\}$ and $\mathcal{T}$.

5. Control scheme

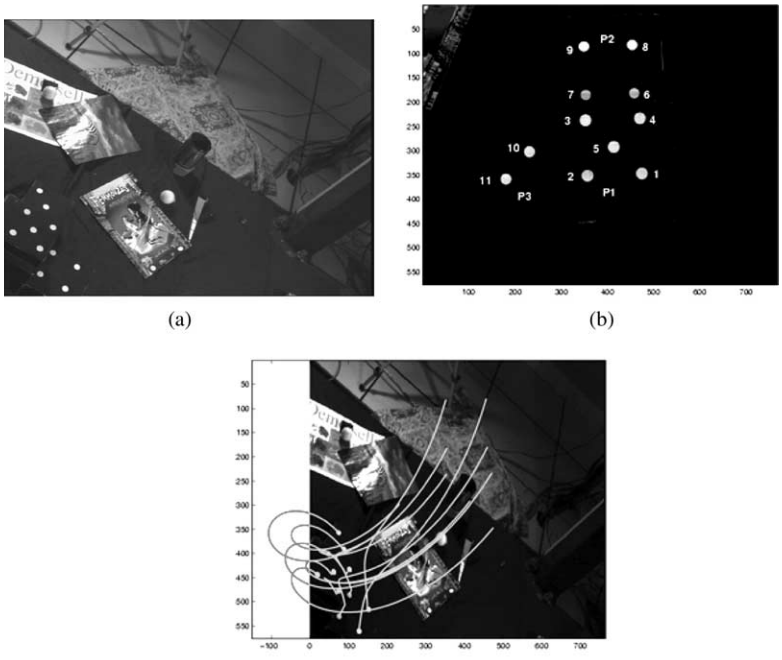
To track the trajectories using an image-based control scheme, we use the task function approach introduced by Samson et al. [17]. A vision-based task function $\mathbf{e}$ to be regulated to $\mathbf{0}$ is defined by [4]:

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

The time varying vector $\mathbf{s}^*(t)$ is the desired trajectory of $\mathbf{s}$ computed as previously explained and the matrix $\hat{\mathbf{L}}^{+}$ is the pseudo-inverse of a chosen model of $\mathbf{L}$. The value of $\mathbf{L}$ at the current desired position is used for $\hat{\mathbf{L}}$. More precisely: $\hat{\mathbf{L}} = \mathbf{L}(\mathbf{s}^*(t), \mathbf{\Gamma}^*(t), \hat{d}^*)$, $\hat{d}^*$ being an estimated value of d^* (see (20)):

$$\mathbf{L}(\mathbf{s}^*(t), \mathbf{\Gamma}^*(t), \hat{d}^*) = \left[\frac{1}{\hat{d}^*}\mathbf{S}(\mathbf{s}^*(t), \mathbf{\Gamma}^*(t)), \mathbf{Q}(\mathbf{s}^*(t)). \right]$$

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),



(a)

(b)

(c)

Fig. 5. (a) Initial image, (b) desired images, and (c) planned trajectories without any repulsive potential.

the corresponding control law is

$$\mathbf{T}_\mathrm{c} = -\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t},$$

where $\mathbf{T}_\mathrm{c}$ is the camera velocity sent to the robot controller. If the target is known to be motionless, we have $(\partial \mathbf{e}/\partial t) = -\hat{\mathbf{L}}^+(\partial \mathbf{s}^*/\partial t)$ and the camera velocity can be rewritten as:

$$\mathbf{T}_\mathrm{c} = -\lambda \mathbf{e} + \hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t},$$

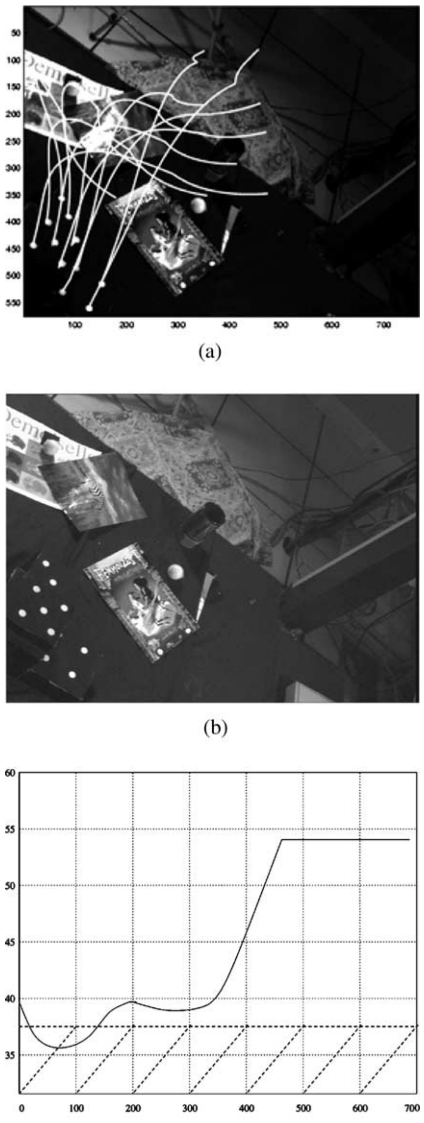
where the term $\hat{\mathbf{L}}^+(\partial \mathbf{s}^*/\partial t)$ allows to compensate the tracking error. More precisely, we have from (23):

$$\frac{\partial \mathbf{s}^*}{\partial t} = 3\mathbf{A}_k t^2 + 2\mathbf{B}_k t + \mathbf{C}_k, \quad (k-1)\Delta T \le t \le k\Delta T.$$

This control law posses nice degrees of robustness with respect to modeling errors and noise perturbations since the error function used as input remains small and is directly computed from visual features.

6. Experimental results

The proposed methods have been tested on a six degree-of-freedom eye-in-hand system. The intrinsic parameters given by the camera manufacturer are used. Since we were not interested in image processing issues in this paper, the target is composed by eleven white marks lying on three different planes (P1, P2 and P3, see Fig. 5). The extracted visual features are the image coordinates of the center of gravity of each mark. The specified visual task consists in a positioning task with respect to an unknown object. The images corresponding to the initial and desired camera positions are given in Figs. 5(a) and (b), respectively. In order to emphasize the importance of the introduced constraints in the trajectories, we first perform the path planning without repulsive potential. We can see that the visual features get out largely of the camera field of view (see Fig. 5(c)). Thus the servoing cannot be realized. In the experiment whose results are given in Fig. 6, only the repulsive potential associated to the self-occlusions avoidance is activated. In that case, the visibility constraint is ensured (see Fig. 6(a)), but point 7 is occluded by plane 1 along the planned trajectories (see Fig. 6(b)). Fig. 6(c) represents the distance d in the image between point 7 and plane 1. In the experiment whose results are given in Fig. 7, the two repulsive potentials are activated and we choose $\zeta = d_{\mathrm{min}} = 37$ in order to avoid the occlusion of point 7 (see Eq. (21)). The object of interest remains in the camera field of view (see Fig. 7(a) and (b)) and mark 7 is not occluded (see Fig. 7(c)). The distance between



(a)

(b)

(c)

Fig. 6. Experiment without repulsive potential associated to the self-occlusions avoidance: (a) planned trajectories, (b) the point 7 is occluded by the plane 1, and (c) distance between point 7 and plane 1.

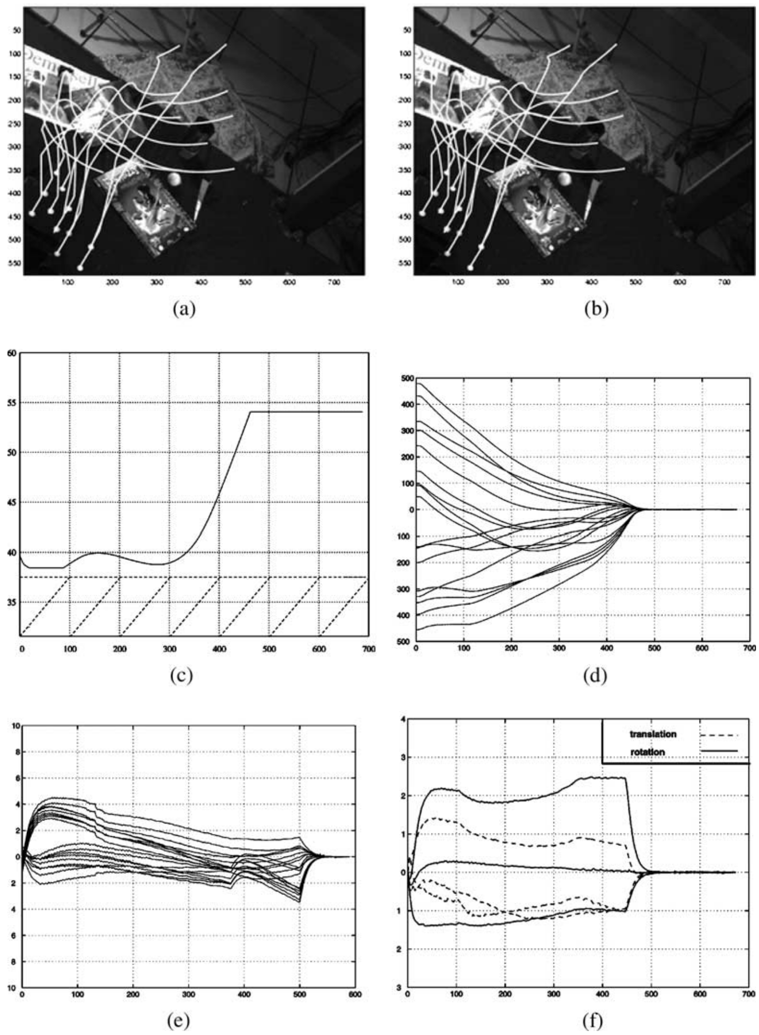


Fig. 7. Both the repulsive potentials are activated: (a) planned trajectories, (b) followed trajectories, (c) distance between point 7 and plane 1, (d) errors in image points coordinates (pixels), (e) tracking errors (pixels), and (f) velocities (cm/s and dg/s).

point 7 in the image and plane 1 is always larger than $d_{\min}$. The servoing step can thus be realized. The error on the coordinates of each target point between its current and its desired location in the image is given in Fig. 7(d). The convergence of the coordinates to their desired value demonstrates the correct realization of the task. The tracking error is plotted in Fig. 7(e). It shows the efficiency of the control scheme (the computed control law is given in Fig. 7(f)) since the maximal error is always less than 5 pixels.

7. Conclusion

In this paper, we have presented a method ensuring the convergence for all initial camera position. By coupling an image-based trajectory generator and an image-based servoing, the proposed method extends the well-known stability of image-based servoing when initial and desired camera location are close to the case where they are distant. The obtained trajectories provide some good expected properties. First, along these trajectories the target remains in the camera field of view and self-occlusions cannot occur. Second the corresponding robot motion is physically realizable and the camera trajectory is a straight line outside the area where the repulsive forces are needed. Future work will be devoted to generate the trajectories in image space of complex features in order to apply our method to complex objects.

References

[1] F. Chaumette, Potential problems of stability and convergence in image-based and position-based visual servoing, in: D. Kriegman, G. Hager, A. Morse (Eds.), The Confluence of Vision and Control, Lecture Notes in Control and Information Science, Vol. 237, Springer, Berlin, 1998, pp. 66–78.

[2] F. Chaumette, E. Marchand, A new redundancy-based iterative scheme for avoiding joint limits: application to visual servoing, in: Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 2, San Francisco, CA, April 2000, pp. 1720–1725.

[3] N.J. Cowan, D.E. Koditschek, Planar image based visual servoing as a navigation problem, in: Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, MI, May 1999, pp. 611–617.

[4] B. Espiau, F. Chaumette, P. Rives, A new approach to visual servoing in robotics, IEEE Transactions on Robotics and Automation 8 (3) (1992) 313–326.

[5] O. Faugeras, F. Lustman, Motion and structure from motion in a piecewise planar environment, International Journal of Pattern Recognition and Artificial Intelligence 2 (3) (1988) 485–508.

[6] 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, Singapore, 1993.

[7] K. Hosoda, K. Sakamoto, M. Asada, Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction, IEEE/RSJ International Conference on Intelligent Robots and Systems 1 (3) (1995) 29–34.

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

[9] O. Khatib, Real time obstacle avoidance for manipulators and mobile robots, International Journal of Robotics Research 5 (1) (1986) 90–98.

[10] J.C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, Dordrecht, Amsterdam, 1991.

[11] E. Malis, 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) (2000) 79–97.

[12] E. Malis, F. Chaumette, S. Boudet, 2 1/2 d visual servoing, IEEE Transactions on Robotics and Automation 15 (2) (1999) 238–250.

[13] Y. Mezouar, F. Chaumette, Path planning in image space for robust visual servoing, in: Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 3, San Francisco, CA, April 2000, pp. 2759–2764.

[14] Y. Mezouar, F. Chaumette, Design and tracking of desirable trajectories in the image space by integrating mechanical and visibility constraints, in: IEEE International Conference on Robotics and Automation, Vol. 1, Seoul, Korea, May 2001, pp. 731–736.

[15] B.J. Nelson, P.K. Khosla, Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance, International Journal of Robotics Research 14 (3) (1995) 255–269.

[16] A. Ruf, R. Horaud, Visual trajectories from uncalibrated stereo, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, September 1997, pp. 83–91.

[17] C. Samson, B. Espiau, M. Le Borgne, Robot Control: The Task Function Approach, Oxford University Press, Oxford, 1991.

[18] R. Singh, R.M. Voyle, D. Littau, N.P. Papanikolopoulos, Alignment of an eye-in-hand system to real objects using virtual images, in: Workshop on Robust Vision for Vision-based Control of Motion, Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 1998.