Path Planning for Robust Image-based Control

Youcef Mezouar and François Chaumette, *Member, IEEE*

Abstract— **Vision feedback control loop techniques are efficient for a large class of applications but they come up against difficulties when the initial and desired robot positions are distant. Classical approaches are based on the regulation to zero of an error function computed from the current measurement and a constant desired one. By using such approach, it is not obvious to introduce any constraint in the realized trajectories and to ensure the convergence for all the initial configurations. In this paper, we propose a new approach to resolve these difficulties by coupling path planning in image space and image-based control. Constraints such that the object remains in the camera field of view or the robot avoids its joint limits can be taken into account at the task planning level. Furthermore, by using this approach, current measurements always remain close to their desired value and a control by image-based servoing ensures the robustness with respect to modeling errors. The proposed method is based on the potential field approach and is applied when object shape and dimensions are known or not, and when the calibration parameters of the camera are well or badly estimated. Finally, real time experimental results using an eye-in-hand robotic system are presented and confirm the validity of our approach.**

Index Terms — **Path planning, Visual servoing**

I. Introduction

CLASSICAL approaches using visual information in feedback control loops are point to point-based, i.e the robot must reach a desired goal configuration starting from a given initial configuration. Needed information is reduced to the goal configuration and a globally stabilizing feedback is required. However, if the initial error is large, such a control may product an erratic behavior and/or large control effort, especially in the presence of modeling errors. Furthermore, keeping all the object in the camera field of view remains a not obvious task. Visual servoing schemes are local feedback control solutions. They thus require the definition of intermediate subgoals in the sensor space at the task planning level. This paper deals with the problem of path planning in image space and presents a complete solution to specify and to track image trajectories of an object observed by an eye-in-hand robotic system.

Position-based and image-based servoing are now well known approaches [22], [7], [9]. The first one [23], [15], based on the computation of a 3-D Cartesian error, requires a perfect CAD-model of the object and a calibrated camera to obtain unbiased pose estimation. The main advantage of this approach is that it controls directly the camera trajectory in the Cartesian space. However, there is no control in the image and some part of the object may get out of the camera field of view during servoing. Additionally, the computation at each iteration of the control loop of a 3-D error may lead to a strong sensitivity with respect to noise perturbations [1].

In the second approach, the control loop is directly closed in the image space. That ensures local convergence and stability in presence of modeling errors and noise perturbations. However, sometimes, and especially when the initial and desired configurations are distant, the camera trajectory induced by image-based servo are neither physically valid nor optimal due to the nonlinearity and potential singularities in the relation from the image space to the workspace [1].

To enlarge the stable region with respect to image-based servo, Malis et *al* propose a globally stabilizing control solution for general setup called *2 1/2 D* visual servoing [14]. However *2 1/2 D* visual servoing techniques are not purely image-based. The robustness of image-based techniques are thus not fully obtained. In particular, the sensitivity to noise measurement, due to the computation of an 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.

The key idea of our work is to use the local stability and robustness of image-based servoing by specifying adequate trajectories to follow in the image. 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 scheme, the error to regulate remains small. A common deficiency of all the previously cited control schemes is the incapacity to introduce constraints such that the whole observed object remains in the camera field of view (visibility) or such that the robot avoids its joint limits during servoing. When the displacement to realize is large, this deficiency often leads to the failure of the servoing process.

The approach proposed in this paper, by coupling a path planning step and a purely image-based control, extends the local robustness and stability of image-based control to large error. Constraints (visibility and joint limits avoidance) are also introduced. Our path planning strategy is based on the potential field method [10], [11] and can be applied even if none 3-D model of the considered object is available, and even if the camera intrinsic parameters are badly estimated. Our scheme consists of three phases:

1. To plan directly trajectories in the image space would require to introduce the constraint that the computed images correspond to physically valid camera positions.

Manuscript received January 16, 2001 ; revised October 2, 2001. This paper was recommended for publication by Associate Editor N. Xi and Editor S. Hutchinson upon evaluation of the reviewers' comments. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Seoul, South Korea, April 2001.

Y. Mezouar was with IRISA Rennes, France. He is now with LASMEA, Campus Universitaire des Cézeaux, 63177 Aubière-cedex, France (e-mail: mezouar@lasmea.univ-bpclermont.fr).
F. Chaumette is with IRISA/INRIA Rennes, 35042 Rennes-cedex, France (e-mail: chaumette@irisa.fr).

To solve this inverse problem seems to be very complex. Similarly, introducing in a pure image planner the constraint that the camera 3D trajectory has a particular form seems to be out of reach. We prefer thus to plan first the camera trajectory and then to deduce the corresponding trajectories in the image space. The discrete geometric camera path is performed as a sequence of N intermediate camera poses which approaches as much as possible a straight line translation. In this phase, the mechanical and visibility constraints are introduced. Contrary to other approaches exploiting the robot redundancy [2], [17], the mechanical and visibility constraints can be ensured even if all the robot degrees of freedom are used to realize the task.

2. In order to use a purely image-based control, the discrete geometric trajectory of the target in the image is then determined from the camera path. The obtained image trajectories are not disturbed by modeling errors as shown in Sections IV-D and IV-E contrarily to the corresponding camera trajectory.

3. In the third phase, continuous and differentiable geometric paths in the image with an associated timing law are generated to improve the control behavior and to manage easily the computation of the reference. These trajectories are tracked efficiently by exploiting the local robustness and stability of a purely image-based control.

Only few papers deal with path planning in image space. For a very simple case (the world is supposed to be planar and the camera is one dimensional), Cowan and Koditschek describe in [3] a globally stabilizing method using navigation function guaranteeing visibility. In [8], a trajectory generator using a stereo system is proposed and applied to obstacle avoidance. An alignment task using intermediate views of an object synthesized by image morphing is presented in [21]. A path planning for a straight-line robot translation observed by a weakly calibrated stereo system, is performed in [19]. However, none of these previous works were dealing with robustness issues. In [16], we have described preliminary results concerning our potential field-based path planning scheme described in this paper.

The paper is organized as follows. In Section II, we recall some basic fundamentals. The method of path planning for a known object is presented in Section III and extended to the case where a 3-D model of the object is not available in Section IV. Camera calibration errors are also considered. In Section V, a timing law is associated to the geometric path. In Section VI, we show how to use an image-based control approach to track the trajectories. The experimental results are given in Section VII.

II. Fundamentals

In this section, we introduce notations and concepts that are necessary to design our path planning generator.

A. General description and notation

Let $\mathcal{F}_o$ be a frame attached to the target, $\mathcal{F}_i$, $\mathcal{F}_k$ and $\mathcal{F}_*$ be the frames attached to the camera in its initial, current and desired positions respectively (see Figure 1).

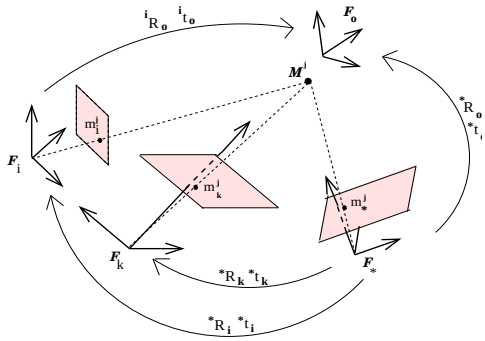


Fig. 1. Computing intermediate views of the target points

$\mathcal{M}^j$ denotes a 3-D target point with homogeneous coordinates $\mathbf{M}_x^j = [\widetilde{\mathbf{M}}_x^{jT}\, 1]^T = [X_x^j\, Y_x^j\, Z_x^j\, 1]^T$ in $\mathcal{F}_x$ (with $x \in \{o, i, k, *\}$). $\mathcal{M}^j$ is projected in the image at time k onto a point with homogeneous normalized coordinates $\mathbf{m}_k^j = [x_k^j\, y_k^j 1]^T$. The corresponding coordinates in pixels are denoted $\mathbf{p}_k^j = [u_k^j\, v_k^j\, 1]^T = [\widetilde{\mathbf{p}}_k^{jT}\, 1]^T = \mathbf{A}\mathbf{m}_k^j$, where the matrix $\mathbf{A}$ is a non singular matrix containing the camera internal parameters. More precisely, we have:

$$\mathbf{A} = \begin{bmatrix} fp_u & -fp_u\cot(\theta) & u_0 \\ 0 & fp_v/\sin(\theta) & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & \alpha_{uv} & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where u_0 and v_0 are the pixels coordinates of principal point, f is the focal length, p_u and p_v are the magnifications respectively in the u and v directions, and θ is the angle between these axes.

The aim of our work is to design a trajectories generator performing a continuous and differentiable curve $\mathbf{s}(t) = [\widetilde{\mathbf{p}}_{(t)}^{1^T} \cdots \widetilde{\mathbf{p}}_{(t)}^{n^T}]^T$ between the initial configuration $\mathbf{s}_i = [\widetilde{\mathbf{p}}_i^{1^T} \cdots \widetilde{\mathbf{p}}_i^{n^T}]^T$ and the desired one $\mathbf{s}_* = [\widetilde{\mathbf{p}}_*^{1^T} \cdots \widetilde{\mathbf{p}}_*^{n^T}]^T$. First, the discrete geometric camera path is performed as a sequence of N intermediate camera poses $\tau = \{\Upsilon_k \;/\; k \in 1 \cdots N\}$ using the potential field strategy described in the next subsection. Then, the discrete object trajectory in the image $\mathcal{S} = \{\mathbf{s}_k \;/\; k \in 1 \cdots N\}$ and the robot trajectory in the joint space $\mathcal{Q} = \{\mathbf{q}_k \;/\; k \in 1 \cdots N\}$ are obtained from τ. Finally, a continuous and differentiable geometric path in the image $\mathbf{s}(t)$ is performed and tracked (see Figure 2).

B. 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 [10], [11].

B.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 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 parameterization of the robot workspace $\mathcal{W} \subset \mathbf{R}^p$,

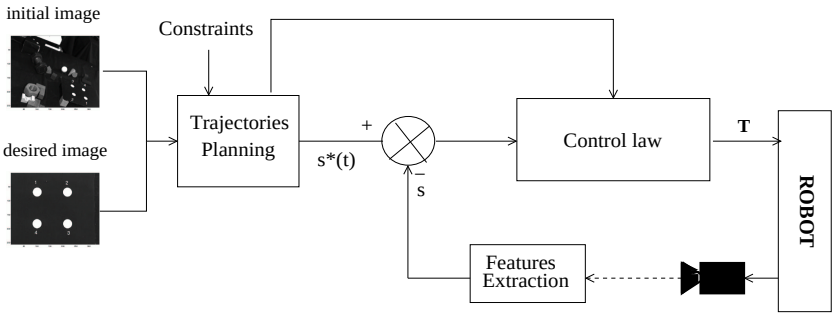


Fig. 2. Block diagram of the proposed method

is induced by the potential function. This force is defined as $\mathbf{F}(\Upsilon) = -\vec{\nabla}_{\Upsilon}^{T} V$ where $\vec{\nabla}_{\Upsilon}^{T} V$ 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, $\mathbf{F}_a(\Upsilon) = -\vec{\nabla}_{\Upsilon}^{T} V_a$ and $\mathbf{F}_r(\Upsilon) = -\vec{\nabla}_{\Upsilon}^{T} V_r$, 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:

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

where k is the increment index and ε_k is a positive scaling factor denoting the length of the kth increment.

B.2 Modified Forces

Consider the unconstrained problem:

$$min\, V(\Upsilon), \quad \Upsilon \in \mathbf{R}^p$$

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

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

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

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

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

$$\dot{\mathbf{f}} = -\epsilon \left(\frac{\partial \mathbf{f}}{\partial \Upsilon}\right) \mathbf{Q} \vec{\nabla}_{\Upsilon}^{T} V = -\epsilon \left(\frac{\partial \mathbf{f}}{\partial \Upsilon}\right) \mathbf{Q} \left(\frac{\partial \mathbf{f}}{\partial \Upsilon}\right)^{T} \vec{\nabla}_{\mathbf{f}}^{T} V \tag{4}$$

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:

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

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

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

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

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

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 follow: 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 joint positions remain between their limits

To deal with the first constraint, the motion is firstly planned in the 3-D Cartesian space and then projected in the image space. The attractive potential (V_{Υ}) pulling the robot toward the goal configuration (Υ_*) is thus defined in the 3D-Cartesian space. The second and the third constraints are introduced through a repulsive potential $V_{\mathbf{s}}$ defined in the image and a repulsive potential $V_{\mathbf{q}}$ defined in the joint space. The total force is given by:

$$\mathbf{F} = \mathbf{F}_{\Upsilon} + \gamma \mathbf{F}_{\mathbf{s}} + \chi \mathbf{F}_{\mathbf{q}} \tag{7}$$

where the scaling factors γ and χ allow us to adjust the relative influence of the different forces. The total force given by the formula (7) can potentially lead to local minima. This is intrinsically due to the path planning strategy used. In practice, we have never encountered configurations leading to local minima and the parameters γ and χ

have been fixed to 1. Note that the defined forces are 6-dimensional vectors and thus to reach a local minima, the 6 components of the total force (that is the pondered sum of the 3 forces) must be equal to zero. Such configurations have a low probability to be attained. A basic strategy to take out of potential local minima is to execute a motion by favoring the repulsive force (i.e by increasing parameters γ and χ). Obviously such strategy makes no formal guarantee to reach the global minimum. In [3], to guarantee such property in the context of visual servo, potential function free of stable local minima (*navigation function* [18]) are constructed. However, constructing such a navigation function requires the complete knowledge of the space topology, and many advantages of the proposed approach in this paper would be lost:robustness with respect to modeling errors, application to object with unknown CAD model. According to (6), the artificial forces can be written as follow:

$$\begin{cases} \mathbf{F}_\Upsilon = -\left(\frac{\partial \Upsilon}{\partial \Upsilon}\right)^+ \vec{\nabla}_\Upsilon^T V_\Upsilon & = & -\vec{\nabla}_\Upsilon^T V_\Upsilon \\ \mathbf{F}_\mathbf{s} = -\left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\frac{\partial \mathbf{r}}{\partial \Upsilon}\right)^+ \vec{\nabla}_\mathbf{s}^T V_\mathbf{s} & = & -\mathbf{M}^+\mathbf{L}^+\vec{\nabla}_\mathbf{s}^T V_\mathbf{s} \\ \mathbf{F}_\mathbf{q} = -\left(\frac{\partial \mathbf{q}}{\partial \mathbf{r}}\frac{\partial \mathbf{r}}{\partial \Upsilon}\right)^+ \vec{\nabla}_\mathbf{q}^T V_\mathbf{q} & = & -\mathbf{M}^+\mathbf{J}(\mathbf{q})\vec{\nabla}_\mathbf{q}^T V_\mathbf{q} \end{cases} \tag{8}$$

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

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

where:

$$\mathbf{a} = \begin{bmatrix} \alpha_u & \alpha_{uv} \\ 0 & \alpha_v \end{bmatrix}$$

When $\mathbf{s}$ is composed of the image coordinates of n points, the corresponding interaction matrix is:

$$\mathbf{L}(\mathbf{s}, \mathbf{Z}) = \left[\mathbf{L}^T(p^1, Z^1) \cdots \mathbf{L}^T(p^n, Z^n)\right]^T$$

III. Path planning for a known target

In this part, the calibration parameters and the 3-D model of the target are supposed to be perfectly known. Knowing the coordinates $\mathbf{M}_o^j$, expressed in $\mathcal{F}_o$, of at least four points $\mathcal{M}^j$, it is possible from their projection to compute the initial and desired poses with respect to $\mathcal{F}_o$ [4], [12], that is the rotation matrix ${}^i\mathbf{R}_o$ (resp. ${}^*\mathbf{R}_o$) and the translation vector ${}^i\mathbf{t}_o$ (resp. ${}^*\mathbf{t}_o$) between $\mathcal{F}_i$ and $\mathcal{F}_o$ (resp. between $\mathcal{F}_*$ and $\mathcal{F}_o$).

A. Camera trajectory in the 3-D Cartesian space

Let ${}^*\mathbf{R}_k$ and ${}^*\mathbf{t}_k$ be the rotational matrix and the translational vector between $\mathcal{F}_k$ and $\mathcal{F}_*$ (see Figure 1). Let $\mathbf{u}_k$ and θ_k be the axis and the rotation angle obtained from ${}^*\mathbf{R}_k$. We choose $\Upsilon_k = [\,{}^*\mathbf{t}_k^T\ (\mathbf{u}\theta)_k^T]^T$ as a parameterization of the workspace. We thus have $\Upsilon_i = [\,{}^*\mathbf{t}_i^T\ (\mathbf{u}\theta)_i^T]^T$ and $\Upsilon_* = \mathbf{0}_{1\times 6}$. The initial camera position Υ_i is obtained from ${}^*\mathbf{t}_i$ and ${}^*\mathbf{R}_i$:

$$\begin{cases} {}^*\mathbf{R}_i & = {}^*\mathbf{R}_o\, {}^i\mathbf{R}_o^T \\ {}^*\mathbf{t}_i & = -\,{}^*\mathbf{R}_i\, {}^i\mathbf{t}_o + {}^*\mathbf{t}_o \end{cases}$$

According to the transition equation (1), where the attractive and the repulsive forces will be given in the subsection III-D and III-E respectively, we construct a path $\tau = \{\Upsilon_k \ /\ k \in 1 \cdots N\}$ as the sequence of successive path segments starting at the initial configuration Υ_i. Furthermore, the Z^j-coordinate in $\mathcal{F}_k$ of each point $\mathcal{M}^j$, which will be used in the repulsive force and in the control law, can be easily obtained at each iteration since:

$$\widetilde{\mathbf{M}}_k^j = [X_k^j\, Y_k^j\, Z_k^j]^T = \left[\ \ {}^*\mathbf{R}_k^T\, {}^*\mathbf{R}_o \quad {}^*\mathbf{R}_k({}^*\mathbf{t}_o - {}^*\mathbf{t}_k)\ \ \right] \mathbf{M}_o^j$$

A complete camera path in the 3-D Cartesian space is obtained as the sequence τ of N intermediate camera poses. A position-based control could thus be used to track it. However, as already explained, it is more interesting to perform features trajectories in the image. Indeed, we thus exploit as well as possible the local stability and robustness of image-based servoing. Furthermore, it avoids the computation of the camera pose at each iteration of the servoing process.

B. Object trajectory in the image space

To perform image-based control, we build the trajectory of the projection $\mathbf{p}^j$ of each point $\mathcal{M}^j$ onto the image using the known coordinates $\mathbf{M}_o^j$ of $\mathcal{M}^j$ in $\mathcal{F}_o$. The trajectory in the image is obtained using the classical assumption that the camera performs a perfect perspective transformation with respect to the camera optical center (pinhole model):

$$r_k^j \mathbf{p}_k^j = [r_k^j u_k^j\ r_k^j v_k^j\ r_k^j]^T = \mathbf{A}[{}^k\mathbf{R}_o\ {}^k\mathbf{t}_o]\mathbf{M}_o^j \tag{9}$$

$\mathbf{p}_k^j$ is easily obtained from (9) by dividing $r_k^j \mathbf{p}_k^j$ by its last component.

C. Trajectories in the joint space

To anticipate the possible encounter of a joint limit and to avoid it, we have to estimate the trajectory of the robot in the joint space. Indeed, the measure of the current joint coordinates is used in the computation of the repulsive potential related to the joint limits avoidance. If the manipulator position in the joint space is represented by $\mathbf{q} = [q^1 \cdots q^m]^T$, we have:

$$\frac{\partial \mathbf{q}}{\partial \Upsilon} = \frac{\partial \mathbf{q}}{\partial \mathbf{r}}\frac{\partial \mathbf{r}}{\partial \Upsilon} = \mathbf{J}^+(\mathbf{q})\mathbf{M} \tag{10}$$

$\mathbf{M}$ is the 6×6 Jacobian matrix that relates the variation of the camera velocity $\mathbf{T}_c$ to the variation of Υ:

$$\mathbf{M} = \begin{bmatrix} {}^*\mathbf{R}_k^T & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{L}_{wk}^+ \end{bmatrix}$$

The computation of $\mathbf{L}_{wk}^{+}$ can be found in [14]:

$$\mathbf{L}_{wk}^{+} = \mathbf{I}_{\mathbf{3}\times\mathbf{3}} + \frac{\theta_k}{2}\mathrm{sinc}^2\left(\frac{\theta_k}{2}\right)[\mathbf{u}_k]_\wedge + (1-\mathrm{sinc}(\theta_k))[\mathbf{u}_k]_\wedge^2$$

where $[\mathbf{u}]_\wedge$ denotes the skew symmetric matrix associated to the vector $\mathbf{u}$. The trajectory of the robot coordinates in the joint space is then obtained from the trajectory of Υ by a linearization of (10) around $\mathbf{q}_k$:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^{+}(\mathbf{q}_k)\,\mathbf{M}\,(\Upsilon_{k+1} - \Upsilon_k)$$

Note that this last equation just leads to an approximation of the joints trajectory. The real one will thus not be exactly the computed one, especially if modeling errors occur.

In the next subsections, we present how the potential functions and the induced forces are defined and calculated.

D. Attractive potential and force

The attractive potential field V_Υ is simply defined as a parabolic function in order to minimize the distance between the current position and the desired one:

$$V_\Upsilon(\Upsilon) = \frac{1}{2}\|\Upsilon - \Upsilon_*\|^2 = \frac{1}{2}\|\Upsilon\|^2$$

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

$$\mathbf{F}_\Upsilon(\Upsilon) = -\vec{\nabla}_\Upsilon^T V_\Upsilon = -\Upsilon \tag{11}$$

When the repulsive potentials are not needed, the transition equation can be written (refer to (1) and (11)):

$$\Upsilon_{k+1} = \left(1 - \frac{\varepsilon_k}{\|\Upsilon_k\|}\right)\Upsilon_k$$

Thus, Υ_k is lying on the straight line passing by Υ_i and Υ_*. As a consequence, the translation performed by the camera is a real straight line since Υ_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 the robot can attain its joint limits along this trajectory. To avoid this potential problems, two repulsive forces are introduced by deviating the camera trajectory when needed.

E. Mechanical and visibility constraints

E.1 Joint limits avoidance

The robot configuration $\mathbf{q}$ is called acceptable if each of its components is sufficiently far away from its corresponding joints limits. That is, $\mathbf{q}$ is acceptable if for all j, $q^j \in [q_{min}^j + l^j;\quad q_{max}^j - l^j]$, q_{min}^j and q_{max}^j being the minimum and the maximum allowable joint values for the jth joint and l^j being the distance of influence of the jth joint limit. We denote $\mathcal{L}$ the subset of the joint space of acceptable configurations. The repulsive potential $V_\mathbf{q}(\mathbf{q})$ is defined as (see Figure 3):

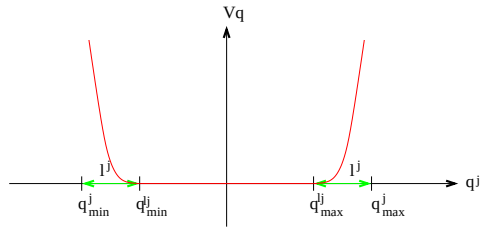
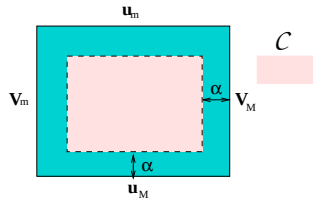
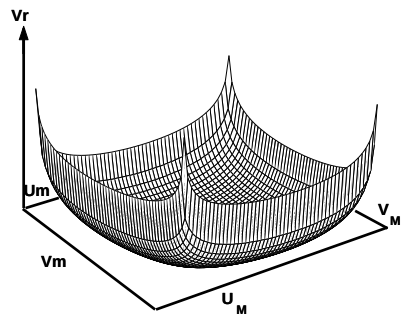


Fig. 3. Repulsive potential for joints limits avoidance



(a)



(b)

Fig. 4. (a) Image limits, (b) Repulsive potential for visibility

$$\begin{cases} -v_q^2 \log\left(\prod_{j=1}^{m}(1-\frac{q^j}{q_{max}^j})(1-\frac{q^j}{q_{min}^j})\right) & \text{if } \mathbf{q} \notin \mathcal{L} \\ 0 & \text{if } \mathbf{q} \in \mathcal{L} \end{cases} \tag{12}$$

In order to obtain a continuous and differentiable potential field $V_\mathbf{q}$, v_q is chosen as a bounded function with null value in the boundary of $\mathcal{L}$:

$$v_q(\mathbf{q}) = \prod_{j=1}^{n}(q^j - q_{max}^{lj})(q^j - q_{min}^{lj})$$

where $q_{min}^{lj} = q_{min}^j + l^j$ and $q_{max}^{lj} = q_{max}^j - l^j$. The potential $V_\mathbf{q}$ is positive or null, tends to infinity as $\mathbf{q}$ gets closer to a joint limit, and it is null when the distance between $\mathbf{q}$ and the robot mechanical limits is more than l^j. The artificial repulsive force deriving from $V_\mathbf{q}$ is (refer to (8)):

$$\mathbf{F}_\mathbf{q} = -\mathbf{M}^{+}\mathbf{J}\vec{\nabla}_\mathbf{q}^T V_\mathbf{q}$$

where $\vec{\nabla}_\mathbf{q}^T V_\mathbf{q}$ is directly obtained from (12).

E.2 Visibility constraint

A point $\mathcal{M}^j$, which projects onto the image plane at a point with image coordinates $\mathbf{p}^j = [u^j\, v^j\, 1]^T$, is known

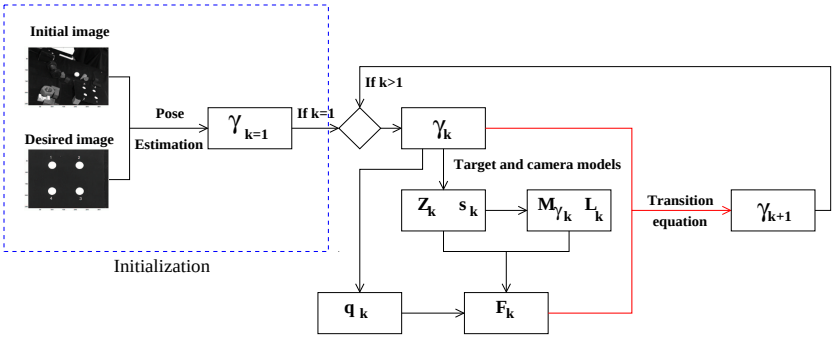


Fig. 5. Block diagram of the path planning for a known object

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 $j \in \{1 \cdots n\}$, $u^j \in [u_m + \alpha;\ u_M - \alpha]$ 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 Figure 4(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 $V_{\mathbf{s}}(\mathbf{s})$ as (see Figure 4(b)):

$$\begin{cases} -v_s^2 \log\Big(\prod_{j=1}^{n}(1-\frac{u^j}{u_M})(1-\frac{u^j}{u_m})(1-\frac{v^j}{v_M})(1-\frac{v^j}{v_m})\Big), & \text{if } \mathbf{s} \notin \mathcal{C} \\ 0, & \text{if } \mathbf{s} \in \mathcal{C} \end{cases} \tag{13}$$

As for the previous potential function, v_s is chosen as a bounded function with null value in the boundary of $\mathcal{C}$:

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

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 $V_{\mathbf{s}}$ 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 $V_{\mathbf{s}}$ is:

$$\mathbf{F}_{\mathbf{s}}(\Upsilon) = -\mathbf{M}^+\mathbf{L}^+\vec{\nabla}_{\mathbf{s}}^T V_{\mathbf{s}} \tag{14}$$

where $\vec{\nabla}_{\mathbf{s}}^T V_{\mathbf{s}}$ is easily obtained from (13).

F. Summary

By using a target model and a calibrated camera, the trajectory of a set of n points in the image has been obtained as a sequence of N vectors $\mathcal{S} = \{\mathbf{s}_k / k \in 1 \cdots N\}$. A block diagram of the path planning scheme is given in Figure 5. The obtained trajectory provides some good expected properties: along this trajectory the target remains in the camera field of view, the corresponding robot motion is physically realizable and the camera trajectory is a straight line outside the area where the repulsive forces are needed. The set $\mathcal{Z} = \{\mathbf{Z}_k = [Z_1 \cdots Z_n] / k \in 1 \cdots N\}$, which will be used in the control law, has been also determined.
In the next part, we extend this method to the case where the object shape and dimensions are unknown and where the calibration is well or badly estimated.

IV. Path planning for an unknown target

In this section, we consider that the target model is not available. In this case the camera pose can not 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 object in the image space. First, we present the method with accurate calibration parameters and then, the robustness with respect to modeling errors is studied. For convenience, the rotation matrix ${}^*\mathbf{R}_k$ and the translation vector ${}^*\mathbf{t}_k$ are denoted $\mathbf{R}_k$ and $\mathbf{t}_k$ in the sequel.

A. Scaled Euclidean reconstruction

Consider a 3-D reference plane Π given in the desired camera frame $\mathcal{F}_*$ by the vector $\pi^T = [\mathbf{n}^* - d^*]$, where $\mathbf{n}^*$ is its unitary normal in $\mathcal{F}_*$ and d^* is the distance from Π to the origin of $\mathcal{F}_*$ (see Figure 6). It is well known that there is a projective homography matrix $\mathbf{G}_k$, relating the image points in the current and the desired images, such that [5]:

$$\alpha_k^j \mathbf{p}_k^j = \mathbf{G}_k \mathbf{p}_*^j + \beta^j \mathbf{e}_k \text{ with } \mathbf{e}_k = -\mathbf{A}\mathbf{R}_k^T \mathbf{t}_k \tag{15}$$

where α_k^j is a positive scaling factor and β^j is a constant scaling factor null if the target point belongs to Π. More precisely, if we define the signed distance $d(\mathcal{M}^j, \Pi) = \pi\ \mathbf{M}_*^j$, we have:

$$\beta^j = -\frac{d(\mathcal{M}^j, \Pi)}{Z_*^j d^*} \tag{16}$$

Given at least four matched points belonging to Π, $\mathbf{G}_k$ can be estimated by solving a linear system. If the plane Π is defined by 3 points, at least five supplementary points are necessary to estimate the homography matrix by using for example the linear algorithm proposed in [13]. Assuming that the camera calibration is known, the Euclidean homography $\mathbf{H}_k$ of plane Π is estimated as follow:

$$\mathbf{H}_k = \mathbf{A}^+\mathbf{G}_k\mathbf{A} \tag{17}$$

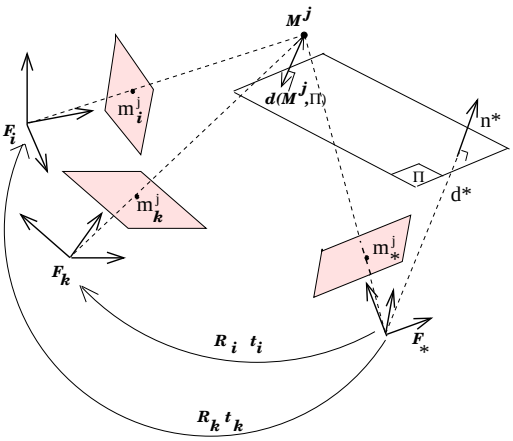


Fig. 6. Scaled 3-D Cartesian trajectory

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

$$\mathbf{H}_k = \mathbf{R}_k^T - \mathbf{R}_k^T \mathbf{t}_{dk} \mathbf{n}^{*T} \quad \text{where} \quad \mathbf{t}_{dk} = \frac{\mathbf{t}_k}{d^*} \tag{18}$$

From $\mathbf{H}_k$, it is thus possible to determine the camera motion parameters (that is the rotation $\mathbf{R}_k$ and the scaled translation $\mathbf{t}_{dk}$) and the structure of the observed scene (for example the vector $\mathbf{n}^*$) [5]. The ratio $\rho_k^j = Z_k^j/d^*$ can easily be estimated [14]:

$$\begin{cases} \rho_k^j = \dfrac{Z_k^j}{d^*} = \dfrac{1 - \mathbf{n}^{*T}\mathbf{R}_k^{2T}\mathbf{t}_{dk}}{\mathbf{n}^{*T}\mathbf{R}_k\mathbf{m}_k^j} & \text{if} \quad \mathcal{M}^j \in \Pi \\ \rho_k^j = \dfrac{Z_k^j}{d^*} = \sigma_k^j \dfrac{\|\mathbf{R}_k^T\mathbf{t}_{dk}\|}{\|\sigma_k^j\mathbf{m}_k^j - \mathbf{R}_k^T\mathbf{m}_*^j\|} & \text{if} \quad \mathcal{M}^j \notin \Pi \end{cases} \tag{19}$$

where $\sigma_k^j = \dfrac{\|[\mathbf{R}_k^T\mathbf{t}_{dk}]_\wedge \mathbf{R}_k^T\mathbf{m}_*^j\|}{\|[\mathbf{R}_k^T\mathbf{t}_{dk}]_\wedge \mathbf{m}_k^j\|}$. These parameters are important since they are used in the path planning generator and in the control scheme.

B. Scaled 3-D Cartesian trajectory

We now choose $\Upsilon_k = [\mathbf{t}_{dk}^T\ (\mathbf{u}\theta)_k^T]^T$ as parameterization of the workspace, since $\mathbf{t}_{dk}$ is available while $\mathbf{t}_k$ is not. We thus have $\Upsilon_i^T = [\mathbf{t}_{di}^T\ (\mathbf{u}\theta)_i^T]$ and $\Upsilon_* = \mathbf{0}_{6\times 1}$. From initial and desired images, it is possible to compute the homography $\mathbf{H}_i$ and then to obtain $\mathbf{R}_i$, $\mathbf{t}_{di}$, $\mathbf{n}^*$ and thus Υ_i. As in the previous section, we construct a path τ starting at Υ_i and oriented along the induced forces given in this case by:

$$\begin{cases} \mathbf{F}_\Upsilon(\Upsilon) &= -\Upsilon \\ \mathbf{F}_\mathbf{s}(\Upsilon) &= -\mathbf{M}^+\mathbf{L}^+\vec{\nabla}_\mathbf{s}^T V_\mathbf{s} \\ \mathbf{F}_\mathbf{q}(\Upsilon) &= -\mathbf{M}^+\mathbf{J}\vec{\nabla}_\mathbf{q}^T V_\mathbf{q} \end{cases}$$

The Jacobian matrix of parameterization $\mathbf{M}(d^*)$ is now given by:

$$\mathbf{M}(d^*) = \begin{bmatrix} d^*\mathbf{R}_k^T & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{L}_{wk}^+ \end{bmatrix} \tag{20}$$

Note that $\mathbf{M}$ now depends of the unknown parameter d^*. However, as shown in the sequel, this parameter does not have any influence on the path planning process if only the visibility constraint is considered. Furthermore, the interaction matrix depends of the depth vector $\mathbf{Z}$. Contrary to the previous case, it can not be computed directly from the scaled parameterization Υ. But the ratio $\rho_k^j = Z_k^j/d^*$ can easily be estimated from the scaled parameterization and the image features according to (19). Thus we rewrite the interaction matrix $\mathbf{L}(\mathbf{s}, \mathbf{Z})$ as follow:

$$\mathbf{L}(\mathbf{s}, \Gamma, d^*) = \begin{bmatrix} \frac{1}{d^*}\mathbf{S} & \mathbf{Q} \end{bmatrix} \tag{21}$$

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

$$\begin{cases} \mathbf{S}^j &= \mathbf{a} \begin{bmatrix} -\frac{1}{\rho_k^j} & 0 & \frac{x_k^j}{\rho_k^j} \\ 0 & -\frac{1}{\rho_k^j} & \frac{y_k^j}{\rho_k^j} \end{bmatrix} \\ \mathbf{Q}^j &= \mathbf{a} \begin{bmatrix} x_k^j y_k^j & -1 - {x_k^j}^2 & y_k^j \\ 1 + {y_k^j}^2 & -x_k^j y_k^j & -x_k^j \end{bmatrix} \end{cases}$$

The partial pose of the camera Υ_k is thus computed at each iteration. Then, $\mathbf{R}_k$ and $\mathbf{t}_{dk}$ are directly obtained from Υ_k. According to (19), the vector $\Gamma = \left[\rho_k^1 \cdots \rho_k^n\right]$ is obtained from Υ_k. Note that in this case the path of the robot in the joint space can be approximated by:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^+(\mathbf{q}_k)\ \mathbf{M}_k(d^*)\ (\Upsilon_{k+1} - \Upsilon_k)$$

Finally, the image features at iteration k are computed as described in the next subsection.

C. Object trajectory in the Image space

The homography matrix $\mathbf{G}_k$ of plane Π relating the current and desired images can be computed from Υ_k by using (17) and (18):

$$\mathbf{G}_k = \mathbf{A}(\mathbf{R}_k^T - \mathbf{R}_k^T\mathbf{t}_{dk}\mathbf{n}^{*T})\mathbf{A}^+ \tag{22}$$

According to (15) the image coordinates of the points $\mathcal{M}^j$ at time k are given by:

$$\mu_k^j\mathbf{p}_k^j = [\mu_k^j u_k^j\ \ \mu_k^j v_k^j\ \ \mu_k^j] = \mathbf{G}_k\mathbf{p}_*^j + \beta^j\mathbf{e}_k \tag{23}$$

where (refer to (15) and (16)):

$$\beta^j\mathbf{e}_k = \frac{d(\mathcal{M}^j, \Pi)}{Z_*^j d^*}\mathbf{A}\mathbf{R}_k^T\mathbf{t}_k = \frac{d(\mathcal{M}^j, \Pi)}{Z_*^j}\mathbf{A}\mathbf{R}_k^T\mathbf{t}_{dk}$$

Using the previous relation, (23) can be rewritten:

$$\mu_k^j\mathbf{p}_k^j = \mathbf{G}_k\mathbf{p}_*^j + \frac{d(\mathcal{M}^j, \Pi)}{Z_*^j}\mathbf{A}\mathbf{R}_k^T\mathbf{t}_{dk} \tag{24}$$

Furthermore, if we apply the relation (24) between the desired and the initial camera positions, we obtain easily:[1]

$$\frac{d(M^j;\ )}{Z^j} = \text{sign}\left(\frac{{}^j_i p^j_i \quad G_i p^j}{(AR_i t_{di})_1}\right)_1 \frac{kG_i p^j \wedge p^j_i k}{kAR_i t_{di} \wedge p^j_i k} \tag{25}$$

The relations (22), (24) and (25) allow to compute ${}^j_k p^j_k$ from $_k$ and the initial and desired visual features. The image coordinates p^j_k are nally computed by dividing ${}^j_k p^j_k$ by its last component.

In the next subsections, we consider that the repulsive force related to the joint limit avoidance is not activated (i.e $_k = 0$) and the e ects of error on the intrinsic parameters and on the depth d are studied.

D. Influence of errors on d

If the joint limits avoidance is not considered, the parameter d appears only in repulsive force through the matrix W de ned as the product of $M^+(d\ )$ and $L^+(s;\ ;d\ )$. However, according to (20) and (21), we have at time k:

$$W(\ _k;\ _k;s_k) = (L(s_k;\ _k;d\ )M_k(d\ ))^+ = \begin{matrix} S(s_k;\ _k)R^T_k\ ^+ \\ Q(s_k)L^+_{wk} \end{matrix}$$

W is independent on the parameter d . Thus, the trajectories in the retinal space are also independent on d and are not a ected by possible errors on d . Let us note that it is unfortunately not the case if joints limits avoidance is also considered.

E. Influence of errors on the intrinsic parameters

If the camera is not perfectly calibrated and $\hat{A}$ is used instead of A , the estimated initial homography matrix is:

$$\hat{H}_i = \hat{A}^+ AH_i A^+ \hat{A} = \ AH_i\ A^+ \tag{26}$$

where $A = \hat{A}^+ A$. Let us make the following assumption:

$$\hat{H}_i = \ AH_i\ A^+ \Rightarrow \hat{H}_k = \ AH_k\ A^+ \tag{27}$$

It means that the initial error on the estimated homography is propagated along the trajectory and implies that the estimated homography at time k can be decomposed in the sum of a matrix similar to a rotation matrix and a rank 1 matrix:

$$\hat{H}_k = \hat{H}_{1\,k} \quad \hat{t}_k \hat{n}^T \tag{28}$$

where $\hat{H}_{1\,k} = \ AR^T_k\ A^+$, $\hat{n}^T = \frac{n^T A}{kn^T A^+ k}$ and $\hat{t}_k = kn^T A^+ k\ AT_k$ with $T_k = R^T_k t_{dk}$ [14]. Obviously, G_i, p^j and p^j_i are not a ected by errors on intrinsic parameters, since they are extracted or computed directly from image data. According to (24), (25) and (28), the obtained homogeneous image coordinates $\hat{p}^j_k$ in the presence of calibration errors are given by:

$${}^j_k p^j_k = AH_k A^+ p^j + \text{sign}\left(\frac{{}^j_i p^j_i \quad G_i p^j}{A\ T_i\ _1}\right)_1 \frac{kG_i p^j \wedge p^j_i k}{kA\ T_i \wedge p^j_i k} A\ T_k \tag{29}$$

[1] $(v)_j$ is the jth components of v.

Since $A^+\hat{t}_k = kn^T A^+ kAT_k$, we have:

$$\begin{cases} \text{sign}\left(\frac{({}^j_i p^j_i \quad G_i p^j)_1}{(A\ T_i)_1}\right) = \text{sign}\left(\frac{({}^j_i p^j_i \quad G_i p^j)_1}{(A\ T_i)_1}\right) \\ \frac{kG_i p^j \wedge p^j_i k}{kA\ T_i \wedge p^j_i k} A^+ \hat{t}_k = \frac{kG_i p^j \wedge p^j_i k}{kA\ T_i \wedge p^j_i k} AT_k \end{cases} \tag{30}$$

Furthermore, we also have:

$$A^+ \hat{H}_k \hat{A}^+ = A^+\ AH_k\ A^+ \hat{A}^+ = AH_k A^+ \tag{31}$$

By injecting (30) and (31) in (29), we nally obtain:

$$\hat{b}^j_k \hat{p}^j_k = \ {}^j_k p^j_k$$

Therefore, under assumption (27), the trajectories in the image are not disturbed by errors on intrinsic parameters. We will check this nice property on the experimental results given in Section VII.

F. Summary

In this part, trajectories of a set of n points in retinal space have been obtained as a sequence of N vectors $S = \{s_k = k\ 2\ 1\ \ N\}$ without any model of the scene. The set $R = \{\ _k = k\ 2\ 1\ \ N\}$, which will be used in the control law, has been also computed. Moreover, we have seen that the planned path is independent of camera calibration if the joint limit avoidance constraint is not considered. A block diagram of the path planning is given in Figure 7.

V. Performing C^2 image trajectories

In the previous subsections, we have obtained discrete trajectories. In order to design continuous and di erentiable curves and thus to improve the dynamic behavior of the system, we use cubic B-spline interpolation. Note that, if the discrete image data is not su cien tly dense, local minima can theoretically be reached on the interpolated trajectory between two successive points (that corresponds to not physically valid camera positions). To deal with this potential problem, the density of the points distribution is chosen su cien tly high. Since this density is chosen by the user by tuning parameter $_k$ in (1), it can be chosen arbitrarily high (typically 500 in the following experiments).

The spline interpolation problem is usually stated as: given data points $S = \{s_k = k\ 2\ 1\ \ N\}$ and a set of parameter values $T = \{t_k = k\ 2\ 1\ \ N\}$, obtain a cubic B-spline curve s(t) such that $s(t_k) = s_k$. In practice, parameter values are rarely given. In our case, we can adjust them to the distribution of the vector of image features s_k or using the distribution of the camera positions $_k$. The distance between two successive camera positions $_k$ and $_{k+1}$ can be chosen constant by xing $"_k$ at a constant value. Let us note that the distance in the image between a point at time k and k+1 is not necessarily constant using a constant value for $"_k$ (see Figure 8). In order to control e cien tly the camera velocity, the time values are chosen spacing proportionally to the distances between camera positions. Thus, the time between two consecutive frames is constant:

paper, the target is composed by white marks (see Figure 9). The extracted visual features are the image coordinates of the center of gravity of each mark.

This section is organized as follows. In Subsection VII-A, we present experiments which confirm that introducing a path planning step improves significantly the robustness with respect to modeling errors. In Subsections VII-B, VII-C and VII-D, only the repulsive potential associated to the visibility constraint is activated. The results obtained with a planar target and an unknown non-planar target are discussed. In Subsection VII-E, we comment experiments obtained when both joint limits avoidance and visibility constraints are taken into account.

A. General interest for path planning

We first show that it is, in general, more interesting to plan the image trajectory $\mathbf{s}^*(t)$ we want to achieve rather than to use a constant reference $\mathbf{s}^*$. To achieve an exponential decay of the sensor signal from its initial value $\mathbf{s}_i$ to its desired one $\mathbf{s}^*$ we use the following classical control law:

$$\mathbf{T}_c = -\lambda \widehat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}^*)$$

where $\widehat{\mathbf{L}} = \mathbf{L}(\mathbf{s}(t), \mathbf{Z}(t))$. With this scheme, we expect that each point moves in the image according to a straight line, even if we know that it is not possible in all cases [1]. In this experiment, we have experimentally chosen a favorable configuration such that each point can move along a straight line in the image. In order to check the robustness with respect to modeling errors, we carried out three experiments:

1. the correct intrinsic parameters are used. The reference $\mathbf{s}^*$ has been taken constant.
2. An error of 40% is introduced on the intrinsic parameters. As in the previous case, the reference $\mathbf{s}^*$ has been taken constant.
3. The same error of 40% is introduced on the intrinsic parameters. A variable reference $\mathbf{s}^*(t)$ is used such that the expected trajectories are straight lines in the image. Let us recall that such trajectories are not always physically valid [1]. Such simple planning approach can thus not be generalized.

The obtained trajectories in the image for the three experiments are given in the Figures 10(a), 10(b) and 10(c) respectively. As expected, the trajectory of each point in the image is a straight line when a correct calibration is used (see Figure 10(a)). But in presence of modeling errors, we note that the trajectories are very far away from the expected ones if a constant value s^* is used (see Figure 10(b)). The motion of the points in the image are thus unpredictable and a part of the image features may get out of the camera field of view. As can be seen in Figure 10(c), by using a variable reference $s^*(t)$, the expected trajectories are really obtained even if important modeling errors are introduced. The motion of each point in the image are thus perfectly predictable. This experiment confirms that a trajectory following in the image space gives better results than a point-to-point motion and improves significantly the robustness with respect to modeling errors.

B. Experimental results for a known planar object

We now consider a planar object with four white marks. The images corresponding to the desired and initial camera positions are given in Figures 11(a) and 11(b) respectively. The corresponding camera displacement between the desired and the initial camera frames is very important (t_x = 300mm, t_y = 550mm, t_z = 120mm $(\mathrm{u}\theta)_x$ = 28dg, $(\mathrm{u}\theta)_y$ = 78dg, $(\mathrm{u}\theta)_z$ =147dg) and, in this case, classical image-based and position-based visual servoing fail. In order to emphasize the importance of the introduced constraint in the trajectories, we perform the path planning of the target without repulsive potential (see Figure 11(c)). We observe that the visual features largely get out of the camera field of view.

In the experiment whose results are reported in Figure 12, the intrinsic parameters given by the camera manufacturer are used. The 3-D parameters used as initialization of the path planning algorithm (that is the initial and desired camera poses) have been obtained by coupling the Dementhon's algorithm [4] and the Lowe's algorithm [12]. The number of intermediate points used is again 500. Planned and tracked trajectories are plotted in Figures 12(a) and 12(b) respectively. We can notice that the tracked trajectories and the planned trajectories are almost similar. This shows the efficiency of our control scheme. The tracking error $(\mathbf{s}(t) - \mathbf{s}^*(t))$ is plotted in Figure 12(e), and it confirms the previous comment since the maximal error is always less than 5 pixels. The error on the coordinates of each target point between its current and its desired location in the image $(\mathbf{s}(t) - \mathbf{s}^*)$ is given in Figure 12(d). The convergence of the coordinates to their desired value demonstrates the correct realization of the task. The computed control law is given in Figure 12(f). We can note its satisfactory variations due to the regularity of the error function. The 3-D camera trajectory is plotted in Figure 12(c).

C. Experimental results for an unknown planar object

We now present the experimental results obtained without taking into account the knowledge of shape and dimension of the object. The initial and desired configurations are the same ones as those used in the previous experiment. The partial motion between the initial and desired camera positions, used to initialize the path planning algorithm is obtained from a projective reconstruction. The number of intermediate points used is 500. As far as calibration is concerned, three different sets of parameters have been used: **1)** *correct calibration:* the correct intrinsic parameters and value of d^* (that is 35 cm) have been used (see Figure 13); **2)** *coarse calibration:* an error of 20% has been added on the intrinsic parameters, while d^* has been set to 20 cm (see Figure 14); **3)** *bad calibration:* an error of 50% has been added on the intrinsic parameters, while d^* has been set to 70 cm (see Figure 15).

1) Correct calibration: First, we note that the trajectories obtained with or without using the object model are almost similar (refer to Figures 12 and 13). Once again, as can be seen in Figures 13(a) and 13(b) the planned and the tracked trajectories are also similar and the tracking error,

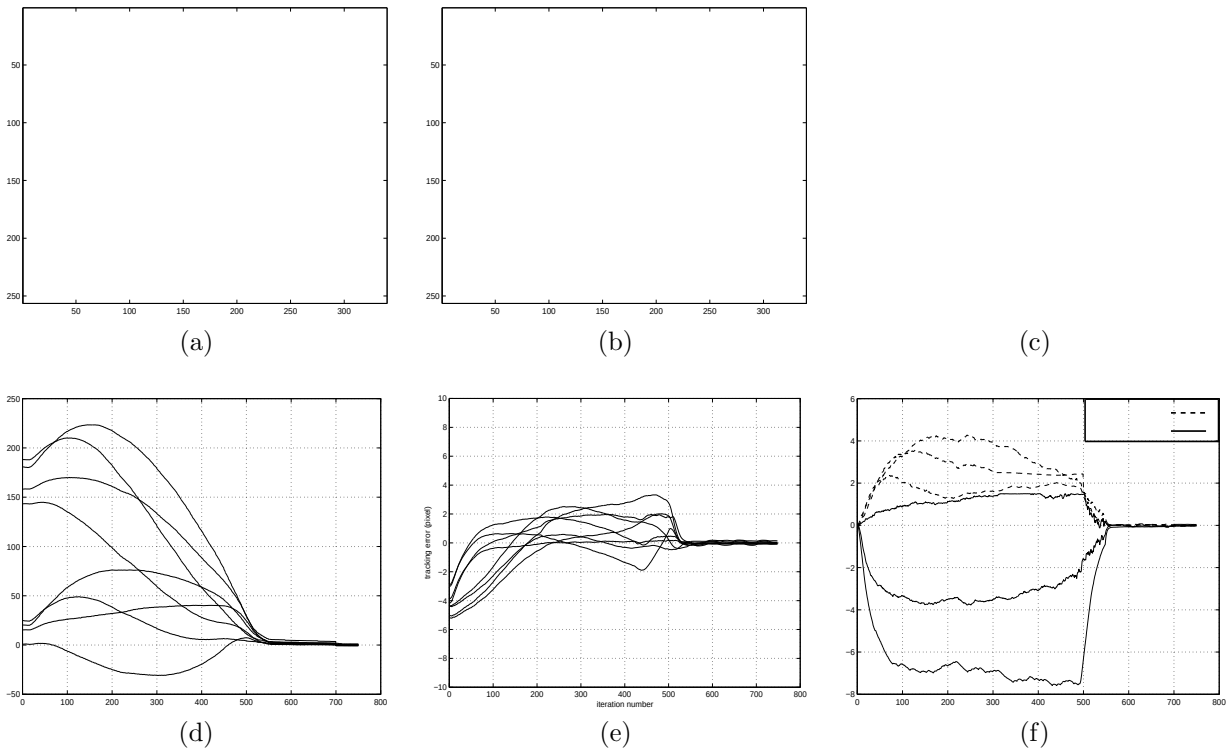


Fig. 13. Same experiment without using the target model and using correct calibration

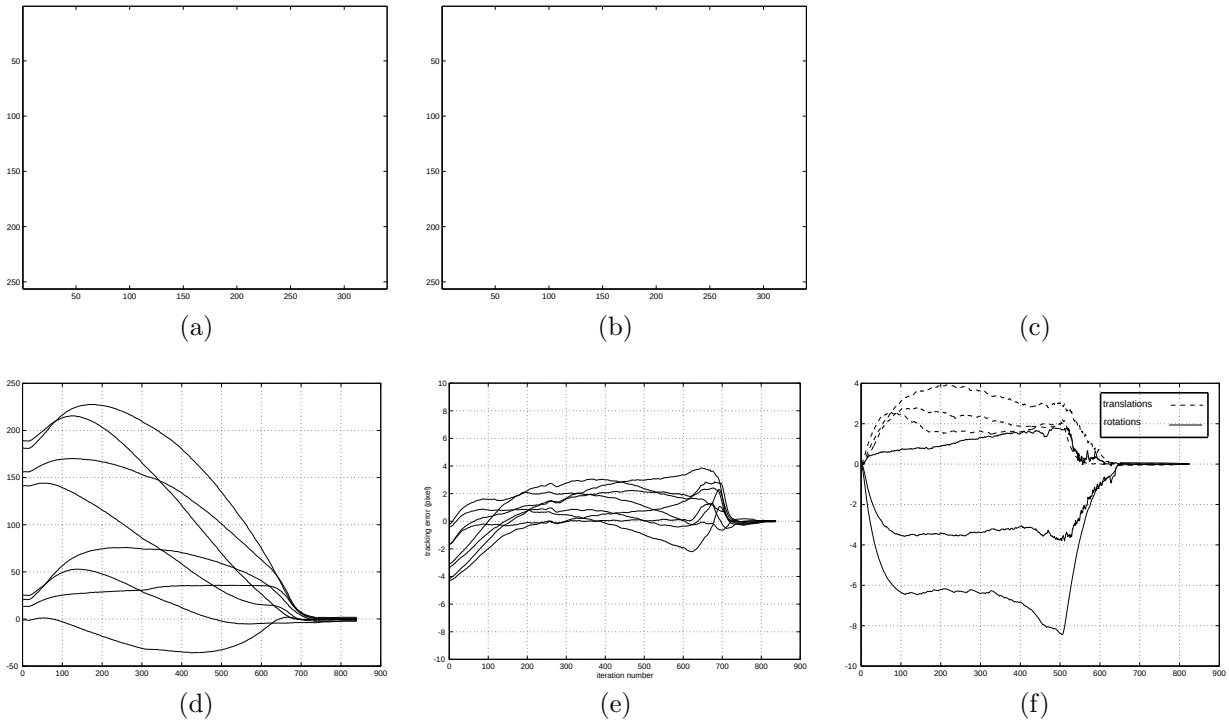


Fig. 14. Same experiment using coarse calibration

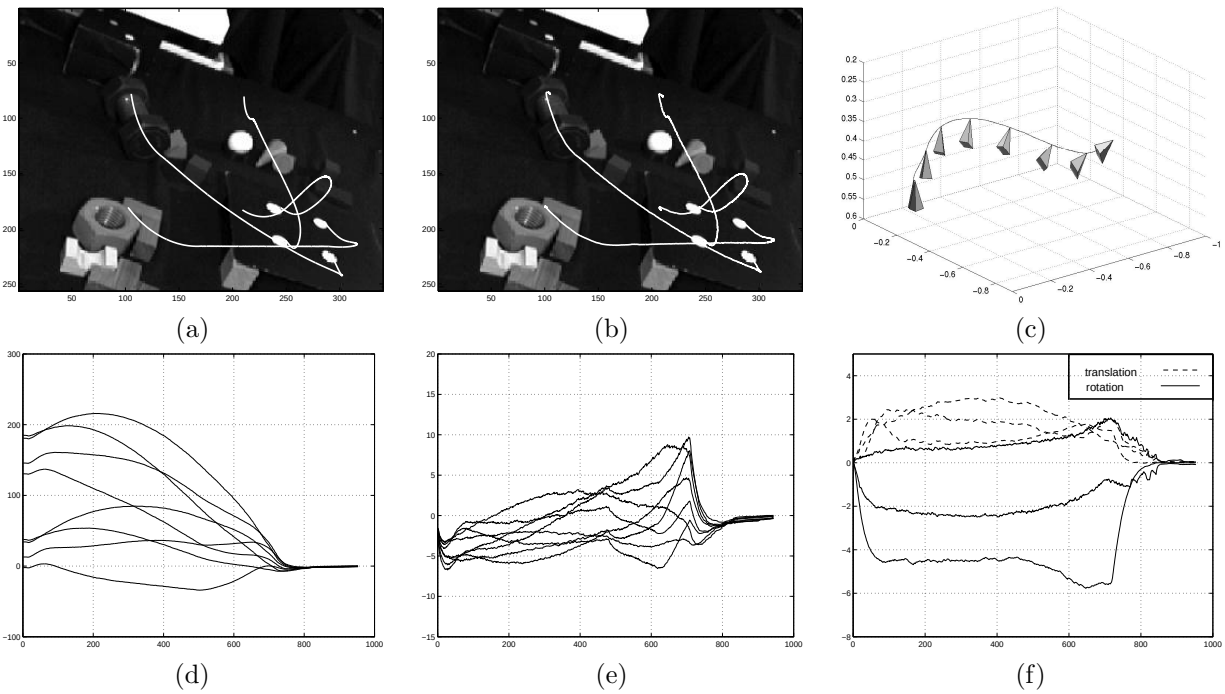


Fig. 15. Same experiment using bad calibration

given in Figure 13(e), remains small during the servoing (less than 5 pixels). Thus, the previous comments, in the case of a known model target, are also valid. The task is correctly realized. This is shown by the convergence of the image points coordinates to their desired value (see Figure 13(d)).

2) and 3) Coarse and Bad Calibrations: We now test the robustness of our approach with respect to calibration errors and errors on the parameter d^*. As can be seen in Figures 13(a), 13(b) and in Figures 14(a), 14(b), the planned and tracked trajectories in the cases of a correct and coarse calibration are similar. The trajectories obtained with bad calibration are close to those obtained in the other cases (see Figure 15). That confirms the robustness of the path planning and of the control scheme with respect to calibration errors and errors on d^*. The tracking error (Figure 14(e)) remains small in the coarse calibration case (less than 5 pixels). It remains satisfactory in the bad calibration case, since always less than 10 pixels (Figure 15(e)). In all the configurations, the tracking error remains sufficiently little to ensure a good behavior of the 2-D control scheme. We note the stability and the robustness of the control law (see Figures 13(f), 14(f) and 15(f)). Finally, we note that the task is correctly realized as well for the coarse calibration case as for the bad calibration case (refer to Figures 14(d) and 15(d)).

D. Experimental results for an unknown non-planar object

The target is now composed of nine white marks lying on three different planes (see Figure 16). In this experiment, the camera displacement is very important ($t_x = -672mm$, $t_y = -1062mm$, $t_z = 468mm$, $(\mathrm{u}\theta)_x = 30.6dg$, $(\mathrm{u}\theta)_y = 56dg$, $(\mathrm{u}\theta)_z = 137dg$). As previously, the path planning algorithm is initialized by using a projective reconstruction and a scaled Euclidean reconstruction. The same values of intrinsic parameters that in the case of a planar object have been used. The depth d^* has been set to 70cm, 50cm and 100cm in the cases of correct, coarse and bad calibration respectively. The number of intermediate points used is now 800 (since the displacement to realize is larger than in the previous experiment). In this part, we only exhibit the results since the comments in the case of an unknown planar object are also valid here. However, we note, once again, that the results are satisfactory in the three cases.

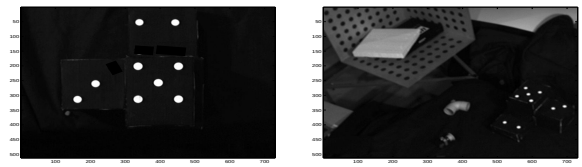


Fig. 16. (a) Initial, (b) desired images of the target

E. Introducing joint limits avoidance

The images corresponding to the desired and initial camera positions are given in Figures 20(a) and 20(b) respectively. On all the following plots, joint positions are normalized between [-1;1], where -1 and 1 represent the joint limits.

We first perform the path planning without repulsive potential. The results are given in Figures 20(c) and 20(d). We can see that the visual features get out largely of the camera field of view and the axis q_5 attains its joint limit. Then, only the repulsive potential associated to the visibility constraint has been activated. In that case, even if the

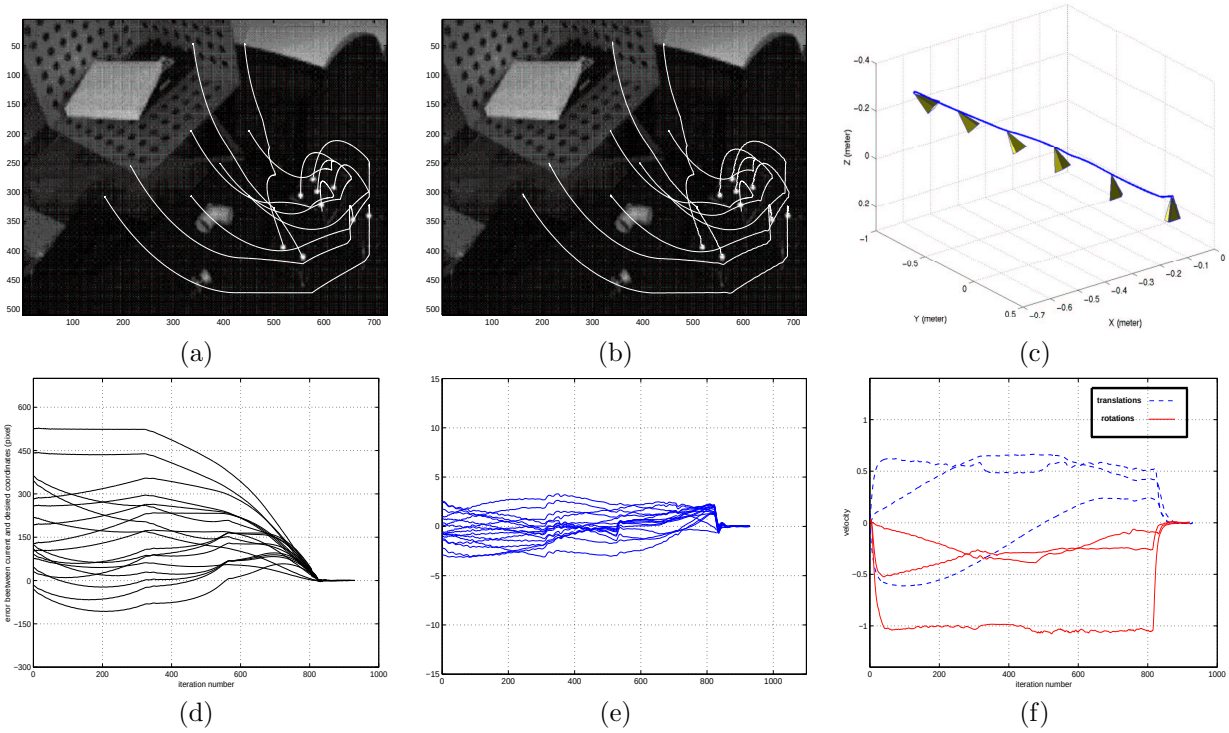


Fig. 17. Experiment with a non-planar object without using the target model and correct calibration: (a) planned trajectories, (b) followed trajectories, (c) camera trajectory, (d) error in image points coordinates (pixels), (e) tracking error (pixels) and (f) velocities (cm/s and dg/s)

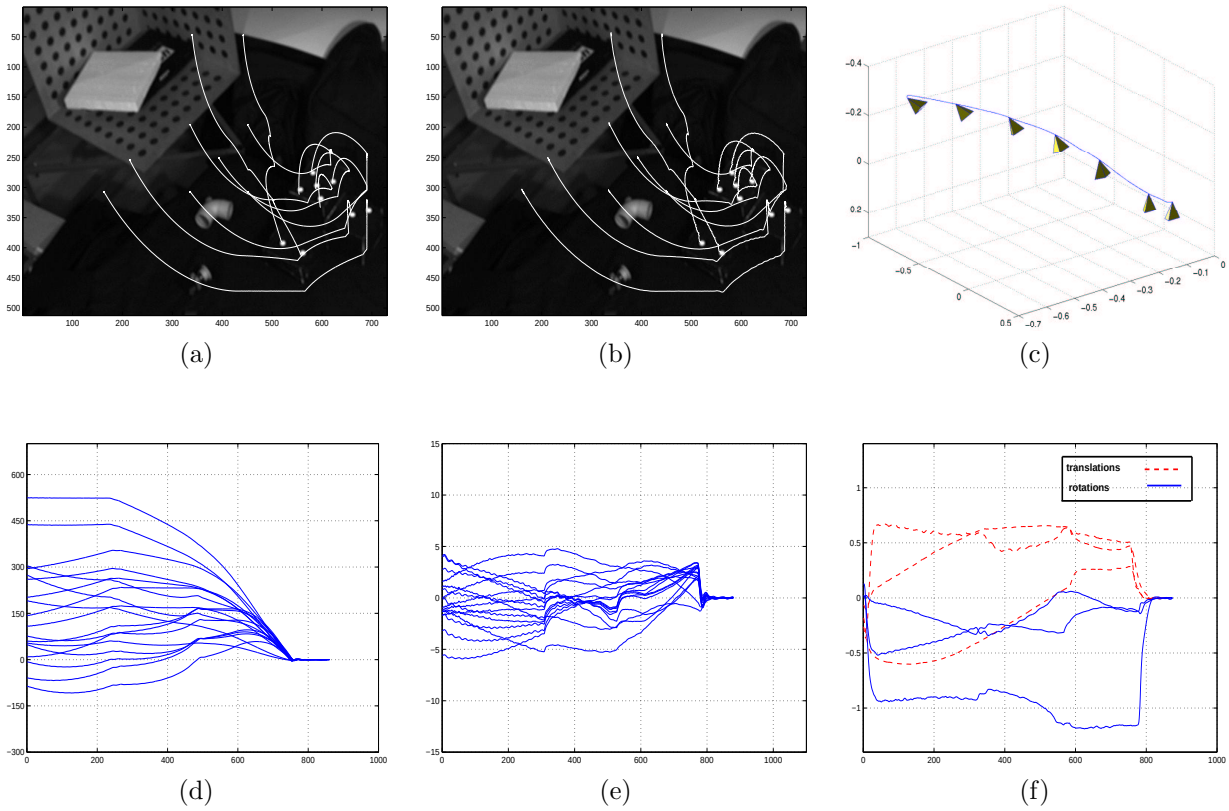


Fig. 18. Same experiment using coarse calibration

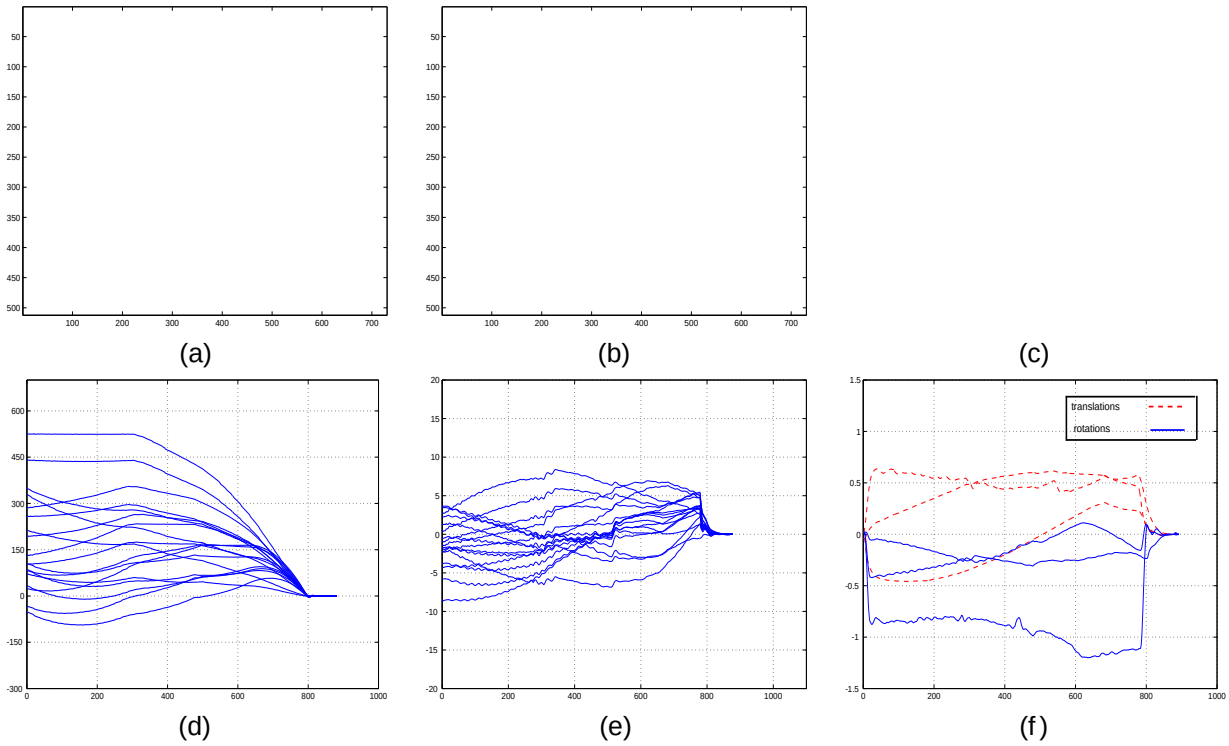


Fig. 19. Same experiment using bad calibration

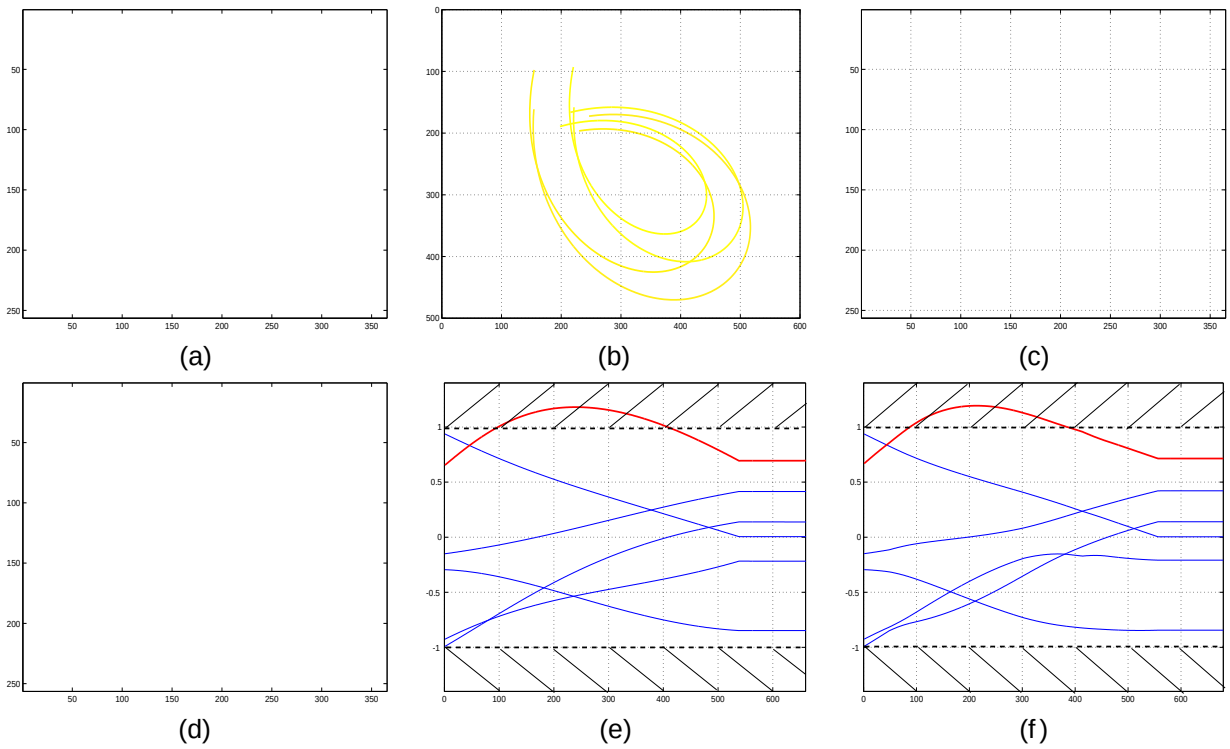


Fig. 20. Initial (a) and desired (d) images; planned trajectories without repulsive potential: (b) in the image, (e) in the joint space; planned trajectories without repulsive potential associated to the joint limits avoidance: (c) in the image, (f) in the joint space