Dealing with visual features occlusions and collisions during a vision-based navigation task in cluttered environments

David FOLIO, Viviane CADENAT
Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse Cedex 9, France
LAAS/CNRS, 7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4, France
E-mail: dfolio@laas.fr,cadenat@laas.fr

Abstract: *In this paper, we address the problem of estimating image features whenever they become unavailable during a vision-based navigation task. The method consists in analytically integrating the relation linking the visual features motion in the image to the 3D camera motion. Simulation results validate our work.*

I. INTRODUCTION

Visual servoing techniques aim at controlling the robot motion using visual features provided by a camera [1, 2]. But, they cannot be used anymore if the image data are lost during the execution of the task. Thus the visual features visibility during a vision-based mission appears as an interesting and challenging problem. Classically, the proposed solutions aim at *avoiding* occlusions and loss. Most of these solutions are dedicated to manipulator arms because they allow to benefit from redundancy to treat this kind of problem [3, 4]. Other techniques preserve visibility by path-planning in the image [5], by acting on specific DOFs [6, 7, 8], by controlling the zoom [9] or by making a tradeoff with the nominal vision-based task [10]. In a mobile robotic context, when executing a vision-based navigation task in a cluttered environment, it is necessary to preserve not only the visual features visibility but also the robot safety. A first answer to this double problem has been proposed in [11, 12]. The developed methods allow to avoid collisions, occlusions and target loss when executing a vision-based task amidst obstacles. However they are restricted to missions where it is possible to avoid both occlusions and collisions without leading to local minima. Therefore, a true extension of these works would be to provide methods which accept that occlusions may effectively occur. A first solution is to allow *some* of the features to appear and disappear temporarily from the image as in [13]. However, this approach is limited to partial occlusions. Another solution which is considered in this paper is to compute the visual features as soon as some or all of them become unavailable. Total visual features loss can then be specifically treated.

The paper is organized as follows. In section II, we model the system and state the problem. In section III, we propose a method allowing to compute the visual features when they become unavailable. Section IV presents the application context, and shows some simulation results validating our work.

II. MODELLING AND PROBLEM STATEMENT

Modelling of the robotic system: We consider the mobile robot SuperScout II[1] equipped with a camera mounted on a pan-platform (see figure 1). It is a small cylindric cart-like vehicle, dedicated to indoor navigation. A DFW-VL500 Sony color digital IEEE1394 camera captures pictures in YUV 4:2:2 format with 640×480 resolution. An image processing module extracts points from the image. The robot is controlled by an on-board laptop computer running under Linux on which is installed a specific control architecture called G^{en}oM (Generator of Module).

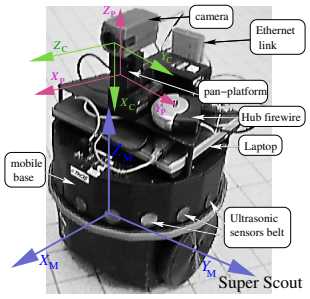

Fig. 1: Nomadic SuperScout II.

First, let us model our system to express the camera kinematic screw. To this aim, consider figure 2. (x,y) are the coordinates of the robot reference point M with respect to the world frame $\mathcal{F}_O$. θ and ϑ are respectively the direction of the vehicle and the pan-platform with respect to $\overrightarrow{\mathbf{x}}_0$ and $\overrightarrow{\mathbf{x}}_M$. P is the pan-platform center of rotation, D_x the distance between M and P. We consider the successive frames:

[1]The mobile robot SuperScout II is provided by the AIP-PRIMECA.

$\mathcal{F}_M\ (M, \vec{\mathbf{x}}_M, \vec{\mathbf{y}}_M, \vec{\mathbf{z}}_M)$ linked to the robot, $\mathcal{F}_P\ (P, \vec{\mathbf{x}}_P, \vec{\mathbf{y}}_P, \vec{\mathbf{z}}_P)$ attached to the pan-platform, and $\mathcal{F}_C\ (C, \vec{\mathbf{x}}_c, \vec{\mathbf{y}}_c, \vec{\mathbf{z}}_c)$ linked to the camera. The control input is defined by the vector $\dot{q} = (v, \omega, \varpi)^T$, where v and ω are the cart linear and angular velocities, and ϖ is the pan-platform angular velocity with respect to $\mathcal{F}_M$. For this specific mechanical system, the kinematic screw $\mathcal{T}^C$ is related to the joint velocity vector by the robot jacobian $\mathbf{J}$: $\mathcal{T}^C = \mathbf{J}\dot{q}$. As the camera is constrained to move horizontally, it is sufficient to consider a reduced kinematic screw $\mathcal{T}_{\mathrm{r}}^C$, and a reduced jacobian matrix $\mathbf{J}_{\mathrm{r}}$ as follows:

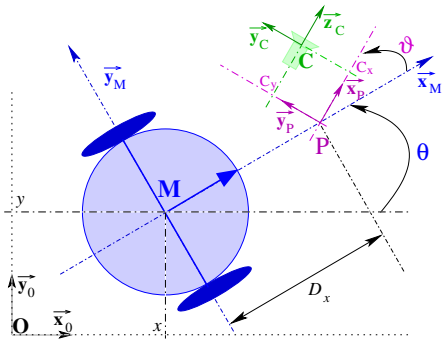

Fig. 2: Modelisation.

$$\mathcal{T}_{\mathrm{r}}^C = \begin{pmatrix} V_{\vec{\mathbf{y}}_C} \\ V_{\vec{\mathbf{z}}_C} \\ \Omega_{\vec{\mathbf{x}}_C} \end{pmatrix} = \begin{pmatrix} -\sin(\vartheta) & D_x\cos(\vartheta)+C_x & C_x \\ \cos(\vartheta) & D_x\sin(\vartheta)-C_y & -C_y \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} = \mathbf{J}_{\mathrm{r}}\dot{q} \tag{1}$$

Problem statement We aim at realizing a vision-based navigation task even if the visual features are temporarily lost during its execution. We propose hereafter a method allowing to compute the visual data when they become unavailable.

III. VISUAL DATA ESTIMATION

In this section, we address the visual data estimation problem. We first introduce some preliminaries and state the estimation problem before presenting our approach.

A. Preliminaries

We consider a static landmark with respect to which is defined the vision-based navigation task. We assume that it can be characterized by n interest points which can be extracted by our image processing. Then, in our case, the set of the considered visual data is represented by a $2n$-dimensional vector s made of the coordinates (U_i, V_i) of each projected point P_i in the image plane (see figure 3). For a fixed landmark, the variation of the visual signals $\dot{s}$ is related to $\mathcal{T}^C$ by means of the interaction matrix $\mathcal{L}$ as shown below [14]:

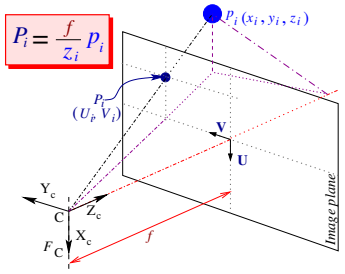

Fig. 3: The pinhole camera model

$$\dot{s} = \mathcal{L}(s, \mathbf{z})\,\mathcal{T}^C = \mathcal{L}(s, \mathbf{z})\,\mathbf{J}_{\mathrm{r}}\dot{q} \tag{2}$$

with $\mathbf{z} = [z_1 \dots z_n]^T$, where z_i is a vector representing the depth of each projected point P_i. This matrix allows to link the visual features motion in the image to the 3D camera motion. It depends mainly on the depth $\mathbf{z}$ and on the considered visual data. In our specific case[2], $\mathcal{L}$ is a $2n \times 3$ matrix deduced from the classical optic flow equations [14] as follows:

$$\mathcal{L}_i(P_i, z_i) = \begin{bmatrix} 0 & \frac{U}{z_i} & \frac{U_i V_i}{f} \\ -\frac{f}{z_i} & \frac{V_i}{z_i} & f + \frac{V_i 2}{f} \end{bmatrix} \qquad \text{where } f \text{ is the camera focal and } i = 1..n \tag{3}$$

B. Estimation Problem Statement

Now, we focus on the problem of estimating (*all* or some) visual data s whenever they become unavailable. Different approaches, such as tracking methods and signal processing techniques, may be used to deal with this kind of problem. Here, we have chosen to use a simpler approach for several reasons. First of all, most tracking algorithms relies on measures from the image which is unavailable in our case. Second, as it is intended to be used to perform complex navigation tasks, the estimated visual signals must be provided sufficiently rapidly with respect to the control law sampling period. Finally, in our application, the initial value of the visual features to be estimated is always known, until the image becomes unavailable. Thus, designing an observer or a filter is not necessary, as this kind of tools is mainly interesting when estimating the state of a dynamic system whose initial value is unknown. Another idea is to use a 3D model of the object together with projective geometry in order to deduce the lacking data. However, this choice would lead to depend on the considered landmark type and would require

[2] Analytical expressions of $\mathcal{L}$ for different kinds of features can be found in [14].

to localize the robot. This was unacceptable for us, as we do not want to make any assumption on the landmark 3D model. Therefore, we have chosen to solve (2) on the base of the visual signals previous measurements and of the control inputs $\dot{q}$.

Now, let us state our problem. As (2) depends on depth $\mathbf{z}$, it is necessary to evaluate this parameter together with the visual data s. Our idea is to express analytically the variation $\mathbf{z}$ with respect to the camera motion. As we consider a target made of n points, we need first to determine the depth variation of each of these points. It can be easily shown that, for one 3D point p of coordinates $(x,y,z)^T$ in $\mathcal{F}_c$ projected into a point $P(U,V)$ in the image plane as shown in figure 3, the depth variation $\dot{z}$ is related to the camera motion according to: $\dot{z} = \mathcal{L}_z \mathcal{T}_r^C = \mathcal{L}_z \mathbf{J}_r \dot{q}$, with $\mathcal{L}_z = [0 \quad -1 \quad \frac{z}{f}V]$. Thus, for the considered task, the system to be solved for one point $P(U,V)$ is given by:

$$\begin{cases} \dot{U} = \frac{U}{z} V_{\overrightarrow{\mathbf{z}}_C} + \frac{UV}{f} \Omega_{\overrightarrow{\mathbf{x}}_C} \\ \dot{V} = -\frac{f}{z} V_{\overrightarrow{\mathbf{y}}_C} + \frac{V}{z} V_{\overrightarrow{\mathbf{z}}_C} + \left(f + \frac{V^2}{f} \right) \Omega_{\overrightarrow{\mathbf{x}}_C} \\ \dot{z} = -V_{\overrightarrow{\mathbf{z}}_C} - \frac{z}{f} V \Omega_{\overrightarrow{\mathbf{x}}_C} \end{cases} \tag{4}$$

Then, introducing $\psi = [U_1 \; V_1 \; \ldots \; U_n \; V_n, \; z_1 \; \ldots \; z_n]^T$, the differential equations to be solved for the considered landmark made of n points are given by: $\dot{\psi} = \begin{pmatrix} \mathcal{L}_s{}^T & \mathcal{L}_{\mathbf{z}}{}^T \end{pmatrix}^T \mathbf{J}_r \dot{q} = \varphi(\psi)$.

C. Analytical resolution

Now let us address the resolution problem. We have already solved the Ordinary Differential Equations (ODE) $\dot{\psi} = \varphi(\psi)$ by using different numerical schemes [15] [16]. In this work, we propose to *analytically* solve $\dot{\psi} = \varphi(\psi)$ by integrating it during the control law sampling period T_{ech}, that is for any $t \in [t_k, t_{k+1}]$. As $\dot{q}$ remains always constant during this time interval, the system to be solved expresses as:

$$\begin{cases} \dot{\psi} & = & \begin{pmatrix} \mathcal{L}_s{}^T & \mathcal{L}_{\mathbf{z}}{}^T \end{pmatrix}^T \mathbf{J}_r(t_k) \, \dot{q}(t_k) \\ \psi(t_k) & = & \psi_k = \left(s_k^T, \mathbf{z}_k^T \right)^T \end{cases} \tag{5}$$

where $\dot{q}(t_k) = \dot{q}_k = (v_k, \omega_k, \varpi_k)^T$. ψ_k is the initial value of ψ, which can be considered known as s_k is directly given by the feature extraction processing (before their loss) and $\mathbf{z}_k$ can be usually characterized from an off-line process. After some computation, we get the following solution for one point $P(U,V)$ (a detailed resolution is available in [17]):

$$U(t) = \frac{z_k U_k}{z(t)} \quad \text{and} \quad V(t) = f \frac{\dot{z} + v_k \cos(\vartheta(t)) + D_x \omega_k \sin(\vartheta(t))) - C_y(\omega_k + \varpi_k)}{z(t)(\omega_k + \varpi_k)} \tag{6}$$

where $z(t)$ is given as shown here:

$$\begin{cases} z(t) = c_1 \sin(A_1(t-t_k)) + c_2 \cos(A_1(t-t_k)) - D_x \cos(\vartheta(t)) + \frac{v_k}{\omega_k} \sin(\vartheta(t)) - C_x & \text{if } \omega_k \neq \varpi_k \text{ and } \omega_k \neq 0 \\ z(t) = -\frac{v_k}{\varpi_k} (\sin(\vartheta(t)) - \sin(\vartheta_k)) + \frac{\omega_k}{\varpi_k} D_x (\cos(\vartheta(t)) - \cos(\vartheta_k)) + z_k & \text{if } \omega_k = -\varpi_k \neq 0 \\ z(t) = c_3 \sin(\varpi_k(t-t_k)) + c_4 \cos(\varpi_k(t-t_k)) - v_k(t-t_k)\cos(\vartheta(t)) + \frac{v_k}{2\varpi_k} \sin(\vartheta(t)) - C_x & \text{if } \omega_k = 0 \text{ and } \varpi_k \neq 0 \\ z(t) = -v_k \cos(\vartheta_k)(t-t_k) + z_k & \text{if } \omega_k = \varpi_k = 0 \end{cases} \tag{7}$$

by considering:

$$\begin{cases} A_1 & = & (\omega_k + \varpi_k) \\ c_1 & = & -\frac{V_k z_k}{f} + D_x \sin(\vartheta_k) + \frac{v_k}{\omega_k} \cos(\vartheta_k) - C_y \\ c_2 & = & z_k + D_x \cos(\vartheta_k) - \frac{v_k}{\omega_k} \sin(\vartheta_k) + C_x \\ c_3 & = & -\frac{V_k z_k}{f} + \frac{v_k}{2\varpi_k} \cos(\vartheta_k) - C_y \\ c_4 & = & z_k - \frac{v_k}{2\varpi_k} \sin(\vartheta_k) + C_x \end{cases}$$

Therefore, the solution requires the determination of $\vartheta(t)$. Thus, knowing the initial value ϑ_k at t_k (usually from embedded encoder), and integrating $\dot{\vartheta} = \varpi$, yields to: $\vartheta(t) = \varpi_k(t-t_k) + \vartheta_k$. Finally, the solution for the set of n points is then given by applying the above resolution on: $\psi = [U_1 \; V_1 \; \ldots \; U_n \; V_n, \; z_1 \; \ldots \; z_n]^T$.

IV. Application

We have chosen to apply our solution in a visual servoing context to compute the visual features when they are lost or unavailable during a navigation task. The considered mission consists in realizing a visually guided navigation task amidst obstacles despite possible occlusions and collisions.

A. The vision-based task to be realized

Our goal is here to position the embedded camera with respect to a landmark made of n points. To this aim, we have applied the visual servoing technique given in [14] to mobile robots as in [18]. In this approach which relies on the task function formalism [19], the visual servoing task is defined as the regulation to zero of the following error function:

$$e_{\rm vs}(q,t) = \mathcal{C}(s(q,t) - s^*) \tag{8}$$

where s^* is the desired value of the visual signal, while $\mathcal{C}$ is a full-rank combination matrix which allows to take into account more visual features than available DOF [14]. Classically, a kinematic controller, $\dot{q}_{\rm vs}$ can be determined by imposing an exponential convergence of $e_{\rm vs}$ to zero: $\dot{e}_{\rm vs} = C\mathcal{L}\mathbf{J}_{\rm r}\dot{q}_{\rm vs} = -\lambda_{\rm vs}e_{\rm vs}$, where $\lambda_{\rm vs}$ is a positive scalar or a positive definite matrix. Fixing $\mathcal{C} = \mathcal{L}^+$ as in [14], we get:

$$\dot{q}_{\rm vs} = \mathbf{J}_{\rm r}^{-1}(-\lambda_{\rm vs})\mathcal{L}^+(s(q,t) - s^*) \tag{9}$$

B. Dealing with occlusions and collisions

Let us recall that our goal is to realize a positioning vision-based task amidst possibly occluding obstacles. Two problems must then be addressed: the visual data loss and the risk of collision. The first one will be treated using the above estimation technique and the second one thanks to a rotative potential field method. We describe the control strategy before presenting the simulation results.

Collision and occlusion detection Our control strategy relies on the detection of the risks of collision and occlusion. The danger of collision is evaluated from the distance $d_{\rm coll}$ and the relative orientation α between the robot and the obstacle deduced from the embedded US sensors. We define three envelopes around each obstacle ξ_+, ξ_0, ξ_-, located at $d_+ > d_0 > d_-$ (see figure 4). We propose to model the *risk of collision* by parameter $\mu_{\rm coll}$ which smoothly increases from 0 when the robot is far from the obstacle ($d_{\rm coll} > d_0$) to 1 when it is close to it ($d_{\rm coll} < d_-$).

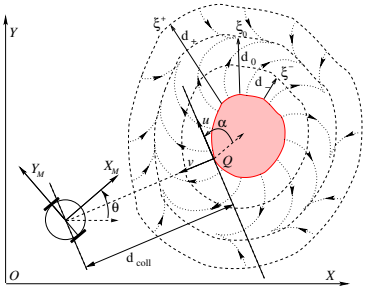

Fig. 4: Obstacle avoidance.

The occlusion risk is evaluated from the detection of the occluding object left and right borders extracted by our image processing algorithm. From them, we can deduce the shortest distance $d_{\rm occ}$ between the image features and the occluding object O, and the distance $d_{\rm bord}$ between O and the opposite image side to the visual features (see figure 5). Defining three envelopes Ξ_+, Ξ_0, Ξ_- around the occluding object located at $D_+ > D_0 > D_-$ from it, we propose to model the risk of occlusion by parameter $\mu_{\rm occ}$ which smoothly increases from 0 when O is far from the visual features ($d_{\rm occ} > D_0$) to 1 when it is close to them ($d_{\rm occ} < D_-$). A possible choice for $\mu_{\rm coll}$ and $\mu_{\rm occ}$ can be found in [11].

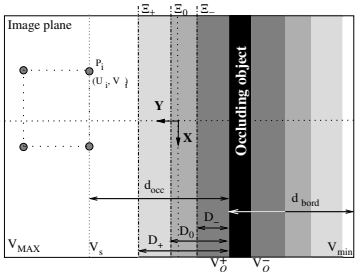

Fig. 5: Occlusion detection.

Global control law design Our global control strategy relies on $\mu_{\rm coll}$ and $\mu_{\rm occ}$. It consists in two steps. First we define two controllers allowing respectively to realize the sole vision-based task and to guarantee non collision while dealing with occlusions in the obstacle vicinity. Second, we switch between these two controllers depending on the risk of occlusion and collision. We propose the following global controller:

$$\dot{q} = (1 - \mu_{\rm coll})\dot{q}_{\rm vs} + \mu_{\rm coll}\dot{q}_{\rm coll} \tag{10}$$

where $\dot{q}_{\rm vs}$ is the previously defined visual servoing controller (9), while $\dot{q}_{\rm coll} = (v_{\rm coll}\ \omega_{\rm coll}\ \varpi_{\rm coll})^T$ handles obstacle avoidance and visual signal estimation if necessary. Thus, when there is no risk of collision, the robot is driven using only $\dot{q}_{\rm vs}$ and executes the vision-based task. When the vehicle enters the obstacle neighborhood, $\mu_{\rm coll}$ increases to reach 1 and the robot moves using only $\dot{q}_{\rm coll}$. This controller is designed so that the vehicle avoids the obstacle while tracking the target, treating the occlusions if any. It is then possible to switch back to the vision-based task once the obstacle is overcome. The avoidance phase ends when both visual servoing and collision avoidance controllers point out the same direction: $sign(\dot{q}_{\rm vs}) = sign(\dot{q}_{\rm coll})$, and if the target is not occluded ($\mu_{\rm occ} = 0$). In this way, we benefit from the avoidance motion to make the occluding object leave the image.

Remark 1 *Controller* (10) *allows to treat occlusions which occur during the avoidance phase. However, obstacles may also occlude the camera field of view without inducing a collision risk. In such cases, we may apply to the robot either another controller allowing to avoid occlusions as done in [11, 12] for instance, or an open-loop scheme based on the computed visual features.*

Obstacle avoidance To design $\dot{q}_{\text{coll}}$, we propose to use a similar approach to the one used in [20]. The idea is to define around each obstacle a rotative potential field so that the repulsive force is orthogonal to the obstacle when the robot is close to it ($d_{\text{coll}} < d_+$), parallel to the obstacle when the vehicle is at a distance d_0 from it, and progressively directed towards the obstacle between d_0 and d^+ (see figure 4). The interest of such a potential is that it can make the robot move around the obstacle without requiring any attractive force, reducing local minima problems. We use the same potential function as in [20]:

$$\begin{cases} U(d_{\text{coll}}) = \frac{1}{2}k_1(\frac{1}{d_{\text{coll}}} - \frac{1}{d^+})2 + \frac{1}{2}k_2(d_{\text{coll}} - d^+)2 & \text{if } d_{\text{coll}} \leq d^+ \\ U(d_{\text{coll}}) = 0 & \text{otherwise} \end{cases} \tag{11}$$

where k_1 and k_2 are positive gains to be chosen. v_{coll} and ω_{coll} are then given by [20]:

$$\dot{q}_{\text{base}} = \begin{pmatrix} v_{\text{coll}} & \omega_{\text{coll}} \end{pmatrix}^T = \begin{pmatrix} k_v F \cos\beta & \frac{k_\omega}{D_x} F \sin\beta \end{pmatrix}^T \tag{12}$$

where $F = -\frac{\partial U}{\partial d_{\text{coll}}}$ is the modulus of the virtual repulsive force and $\beta = \alpha - \frac{\pi}{2d_0} d_{\text{coll}} + \frac{\pi}{2}$ its direction with respect to $\mathcal{F}_M$. k_v and k_ω are positive gains to be chosen. Equation (12) drives only the mobile base in the obstacle neighborhood. However, if the pan-platform remains uncontrolled, it will be impossible to switch back to the execution of the vision-based task at the end of the avoidance phase. Therefore, we have to address the ϖ_{coll} design problem. Two cases may occur in the obstacle vicinity: either the visual data are available or not. In the first case, the proposed approach is similar to [20] and the pan-platform is controlled to compensate the avoidance motion while centering the target in the image. As the camera is constrained to move within an horizontal plane, it is sufficient to regulate to zero the error $e_{\text{gc}} = V_{\text{gc}} - V^*_{\text{gc}}$ where V_{gc} and $V^\star_{\text{gc}}$ are the current and desired ordinates of the target gravity center. Rewriting equation (1) as $\mathcal{T}^C_{\text{r}} = J_{\text{base}}\dot{q}_{\text{base}} + J_\varpi \varpi_{\text{coll}}$ and imposing an exponential decrease to regulate e_{gc} to zero ($\dot{e}_{\text{gc}} = \mathcal{L}_{V_{\text{gc}}}\mathcal{T}^C_{\text{r}} = -\lambda_{\text{gc}} e_{\text{gc}}$, $\lambda_{\text{gc}} > 0$), we finally obtain (see [20] for more details):

$$\varpi_{\text{coll}} = \frac{-1}{\mathcal{L}_{V_{\text{gc}}} J_\varpi}(\lambda_{\text{gc}} e_{\text{gc}} + \mathcal{L}_{V_{\text{gc}}} J_{\text{base}} \dot{q}_{\text{base}}) \tag{13}$$

where $\mathcal{L}_{V_{\text{gc}}}$ is the 2nd row of $\mathcal{L}_i$ evaluated for V_{gc} (see equation (3)). However, if the obstacle occludes the camera field of view, s is no more available and the pan-platform cannot be controlled anymore using (13). At this time, we compute the visual features by integrating the ODE (5) using the analytical solution (6). It is then possible to keep on executing the previous task e_{gc}, even an occlusion occurs. The pan-platform controller during an occlusion phase will then be deduced by replacing the real target gravity center ordinate V_{gc} by the computed one $\widetilde{V}_{\text{gc}}$ in (13). We get:

$$\widetilde{\varpi}_{\text{coll}} = \frac{-1}{\widetilde{\mathcal{L}}_{V_{\text{gc}}} J_\varpi}(\lambda_{\text{gc}} \widetilde{e}_{\text{gc}} + \widetilde{\mathcal{L}}_{V_{\text{gc}}} J_{\text{base}} \dot{q}_{\text{base}}), \tag{14}$$

where $\widetilde{e}_{\text{gc}} = \widetilde{V}_{\text{gc}} - V^*_{\text{gc}}$ and $\widetilde{\mathcal{L}}_{V_{\text{gc}}}$ is deduced from (3). Now, it remains to apply the suitable controller to the pan-platform depending on the context. Recalling that parameter $\mu_{\text{occ}} \in [0;1]$ allows to detect occlusions, we propose the following avoidance controller:

$$\dot{q}_{\text{coll}} = \begin{pmatrix} v_{\text{coll}}, & \omega_{\text{coll}}, & (1-\mu_{\text{occ}})\varpi_{\text{coll}} + \mu_{\text{occ}}\widetilde{\varpi}_{\text{coll}} \end{pmatrix}^T \tag{15}$$

C. Simulation results

The proposed method has been simulated using Matlab software. We aim at positioning the camera with respect to a given landmark despite two obstacles. D_-, D_0 and D_+ have been fixed to 40, 60 and 115 pixels, and d_-, d_0, d_+ to 0.3m, 0.4m, and 0.5m. Figures 6 shows the obtained simulation results. The task is perfectly performed despite the wall and the circular obstacle. At the beginning of the task, there is no risk of collision, nor occlusion, and the robot is driven by $\dot{q}_{\text{VS}}$. When it enters the wall neighborhood, μ_{coll} increases and $\dot{q}_{\text{coll}}$ is applied to the robot which follows the security envelope ξ_0 while centering the landmark. When the circular obstacle enters the camera field of view,

μ_{occ} increases and the pan-platform control smoothly switches from ϖ_{ob} to $\widetilde{\varpi}_{ob}$. It is then possible to move along the security envelope ξ_0 while tracking a "virtual" target until the end of the occlusion. When there is no more danger, the control switches back to $\dot{q}_{VS}$ and the robot perfectly realizes the desired task. Finally, let us notice that the error between the measured visual data s and the estimated ones $\widetilde{s}$ remains unsignificant: about 10^{-10} which correspond to the roundoff error.

V. CONCLUSION AND FUTURE WORKS

In this paper, we have presented a new approach to determine the visual features whenever unavailable during a vision-based task. The proposed method has been validated in simulation to execute a complex vision-based navigation task which cannot be executed if occlusions are not tolerated and collisions avoided. However, this method is restricted to a camera moving in an horizontal plane and to landmarks which can be characterized with points. Therefore, we aim at extending this work to consider other kinds of image features and more complex motions for the camera.

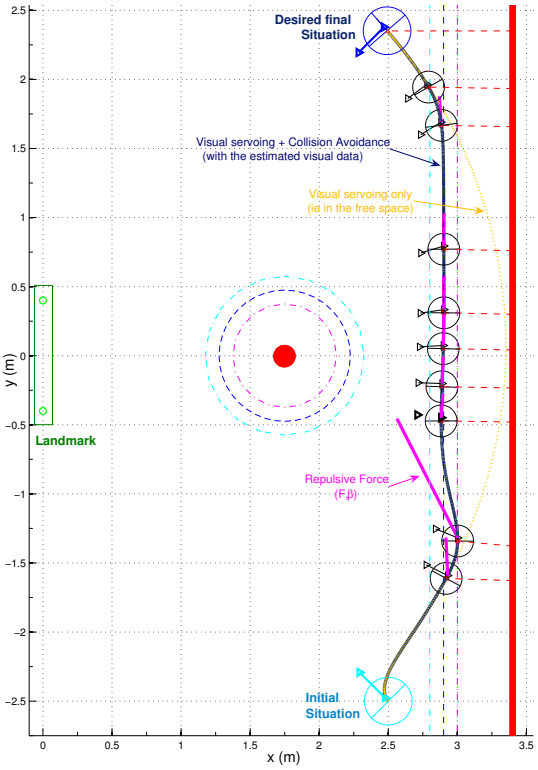

Fig. 6: Robot trajectory.

REFERENCES

[1] P. Corke, *Visual control of robots : High performance visual servoing*. Research Studies Press LTD, 1996.

[2] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *Trans. on Robotics and Automation*, Oct. 1996.

[3] E. Marchand and G. Hager, "Dynamic sensor planning in visual servoing," in *Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998.

[4] N. Mansard and F. Chaumette, "A new redundancy formalism for avoidance in visual servoing," in *Int. Conf. on Intelligent Robots and Systems*, vol. 2, Edmonton, Canada, August 2005, pp. 1694–1700.

[5] Y. Mezouar and F. Chaumette, "Avoiding self-occlusions and preserving visibility by path planning in the image," *Robotics and Autonomous Systems*, Nov. 2002.

[6] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *Trans. on Robotics and Automation*, vol. 17, pp. 507–515, Aug. 2001.

[7] E. Malis, F. Chaumette, and S. Boudet, "2 1/2d visual servoing," *Trans. on Robotics and Automation*, vol. 15, pp. 238–250, Apr. 1999.

[8] V. Kyrki, D. Kragic, and H. Christensen, "New shortest-path approaches to visual servoing," in *Int. Conf. on Intelligent Robots and Systems*, Oct. 2004.

[9] S. Benhimane and E. Malis, "Vision-based control with respect to planar and non-planar objects using a zooming camera," in *Int. Conf. on Advanced Robotics*, Coimbra, Portugal, July 2003.

[10] A. Remazeilles, N. Mansard, and F. Chaumette, "Qualitative visual servoing: application to the visibility constraint," in *Int. Conf. on Intelligent Robots and Systems*, Beijing, China, Oct. 2006.

[11] D. Folio and V. Cadenat, "A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment," in *European Control Conference(ECC-CDC'05)*, Dec. 2005.

[12] ——, "Using redundancy to avoid simultaneously occlusions and collisions while performing a vision-based task amidst obstacles," in *European Conference on Mobile Robots*, Ancona, Italy, Sept. 2005.

[13] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *Trans. on Robotics and Automation*, vol. 21, Dec. 2005.

[14] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *Trans. on Robotics and Automation*, June 1992.

[15] D. Folio and V. Cadenat, "A new controller to perform safe vision-based navigation tasks amidst possibly occluding obstacles," in *ECC*, July 2007.

[16] ——, "Using simple numerical schemes to compute visual features whenever unavailable," in *ICINCO*, May 2007.

[17] D. Folio, "Stratégies de commande référencées multi-capteurs et gestion de la perte du signal visuel pour la navigation d'un robot mobile," Ph.D. dissertation, Université Paul Sabatier, Toulouse, France, July 2007.

[18] R. Pissard-Gibollet and P. Rives, "Applying visual servoing techniques to control a mobile hand-eye system," in *Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995.

[19] C. Samson, M. Leborgne, and B. Espiau, *Robot Control. The Task Function Approach*, ser. Oxford Engineering Series. Oxford University Press, 1991, vol. 22.

[20] V. Cadenat, P. Souères, R. Swain, and M. Devy, "A controller to perform a visually guided tracking task in a cluttered environment," in *Int. Conf. on Intelligent Robots and Systems*, Korea, Oct. 1999.