A sensor-based controller able to treat total image loss and to guarantee non-collision during a vision-based navigation task

David FOLIO and Viviane CADENAT

Abstract— **This paper deal with the problem of executing a vision-based task in an unknown environment. During such a task, two unexpected events may occur: the image data loss due to a camera occlusion and the robot collision with obstacles. We first propose a method allowing to compute the visual data when they are totally lost, before addressing the obstacle avoidance problem. Then, we design a sensor-based control strategy to perform safely vision-based tasks despite complete loss of the image. Simulation and experimental results validate our work.**

I. INTRODUCTION

Visual servoing techniques aim at controlling the robot motion using visual features provided by a camera [1]. Therefore, these techniques cannot be used anymore if these features are lost during the robotic task. A first way to deal with this kind of problem is to use methods allowing to preserve the visual data visibility [2][3][4][5]. However, these works are mostly applied to manipulators. The problem is slightly different in mobile robotics. Indeed, when a robot navigates in an unknown environment using a camera, it is necessary to guarantee both image features visibility and robot safety. Some works in this area propose to fulfill simultaneously these two objectives [6][7]. However, these methods appear to be limited because they do not allow to perform robotic tasks whose success depends on the ability of tolerating the visual data loss.

In this paper, we consider the problem of executing a vision-based task in an unknown environment. During such a task, obstacles may occlude totally the camera or induce a risk of collision. Our first goal is to build a method allowing to compute the image data when they are **entirely unavailable**. Then, we address the obstacle avoidance problem using potential field techniques. Finally, thanks to these results, we design a multi-sensor-based control law able to tolerate the image total loss and guarantee non collision.

The article is organized as follows. Section II is dedicated to the problem statement and to our estimation method. Section III addresses the avoidance problem and describes the chosen control strategy. Finally, last section presents our robotic system and some successful experimental results.

II. VISUAL DATA ESTIMATION

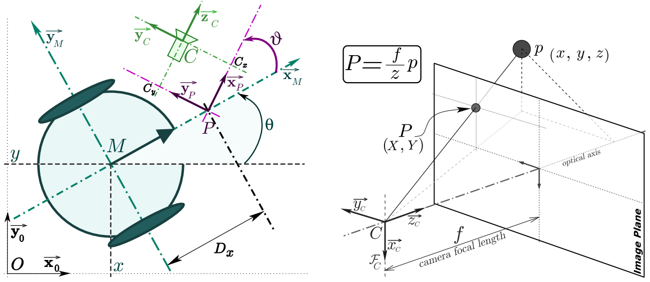
In this section, we focus on the visual data estimation problem. Obstacle avoidance will be treated in next section.

A. Modelling and Preliminaries

1) System modelling: Let us consider figure 1(a). We define the successive frames: $\mathcal{F}_M$ $(M, \overrightarrow{\mathbf{x}}_M, \overrightarrow{\mathbf{y}}_M, \overrightarrow{\mathbf{z}}_M)$ linked to the robot, $\mathcal{F}_P$ $(P, \overrightarrow{\mathbf{x}}_P, \overrightarrow{\mathbf{y}}_P, \overrightarrow{\mathbf{z}}_P)$ attached to the pan-platform, and $\mathcal{F}_C$ $(C, \overrightarrow{\mathbf{x}}_c, \overrightarrow{\mathbf{y}}_c, \overrightarrow{\mathbf{z}}_c)$ linked to the camera. Let ϑ be the direction of the pan-platform wrt. $\overrightarrow{\mathbf{x}}_M$, P the pan-platform center of rotation and D_x the distance between the robot reference point M and P. The control input is defined by: $\dot{\mathbf{q}} = (v, \omega, \varpi)^T$, where v and ω are the cart linear and angular velocities, and ϖ is the pan-platform angular velocity wrt. $\mathcal{F}_M$. For this specific mechanical system, the kinematic screw $\mathbf{v}^c$ is related to the control input by the robot jacobian $\mathbf{J}$: $\mathbf{v}^c = \mathbf{J}\dot{\mathbf{q}}$. As the camera is constrained to move horizontally, it is sufficient to consider a reduced kinematic screw $\mathbf{v}_\mathrm{r}^c = (V_{\overrightarrow{\mathbf{y}}_C}, V_{\overrightarrow{\mathbf{z}}_C}, \Omega_{\overrightarrow{\mathbf{x}}_C})^T$, and a reduced jacobian matrix $\mathbf{J}_\mathrm{r}$ as follows:

$$\mathbf{v}_\mathrm{r}^c = \begin{pmatrix} -\sin(\vartheta) & D_x\cos(\vartheta) + C_x & C_x \\ \cos(\vartheta) & D_x\sin(\vartheta) - C_y & -C_y \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} = \mathbf{J}_\mathrm{r}\,\dot{\mathbf{q}} \qquad (1)$$

where C_x and C_y are the coordinates of C along axes $\overrightarrow{\mathbf{x}}_P$ and $\overrightarrow{\mathbf{y}}_P$ (see figure 1(a)).


$\overrightarrow{\mathbf{y}}_M$
$\overrightarrow{\mathbf{z}}_C$
$\overrightarrow{\mathbf{y}}_C$
C
ϑ
C_x
$\overrightarrow{\mathbf{x}}_M$
C_y
$\overrightarrow{\mathbf{y}}_P$
$\overrightarrow{\mathbf{x}}_P$
P
θ
M
y
$\overrightarrow{\mathbf{y}}_0$
O
$\overrightarrow{\mathbf{x}}_0$
x
D_x
$P = \frac{f}{z} p$
p (x, y, z)
P
(X, Y)
optical axis
$\overrightarrow{y}_C$
$\overrightarrow{z}_C$
C
$\overrightarrow{x}_C$
$\mathcal{F}_C$
f
camera focal length
Image Plane

(a) The robot model.

(b) The pinhole camera model.

Fig. 1. The systems modelling.

2) Preliminaries: Our goal is to realize a vision-based navigation task with respect to a given static landmark despite possible total occlusions. We assume that this landmark can be characterized by n interest points which can be extracted by our image processing. Therefore, the visual data are represented by a $2n$-dimensional vector $\boldsymbol{s}$ made of the coordinates (X, Y) of each projected point P in the image plane as shown on figure 1(b). For a fixed landmark, the variation of the visual signals $\dot{\boldsymbol{s}}$ is related to the reduced camera kinematic screw $\mathbf{v}_\mathrm{r}^c$ thanks to the interaction matrix $\mathbf{L}_{(\boldsymbol{s})}$ as shown below [8]:

$$\dot{\boldsymbol{s}} = \mathbf{L}_{(\boldsymbol{s})}\,\mathbf{v}_\mathrm{r}^c = \mathbf{L}_{(\boldsymbol{s})}\,\mathbf{J}_\mathrm{r}\,\dot{\mathbf{q}} \qquad (2)$$

Authors are with LAAS-CNRS; Université de Toulouse; 7, avenue du Colonel Roche, F-31077 Toulouse, France. and Université de Toulouse; UPS; Email: {dfolio,cadenat}@laas.fr

In the case of n points, $\mathbf{L}_{(s)}= [\mathbf{L}_{(P_1)}{}^T, \ldots, \mathbf{L}_{(P_n)}{}^T]^T$ where $\mathbf{L}_{(P)}$ is classically given by [8]:

$$\mathbf{L}_{(P)} = \begin{pmatrix} \mathbf{L}_{(X,z)} \\ \mathbf{L}_{(Y,z)} \end{pmatrix} = \begin{pmatrix} 0 & \frac{X}{z} & \frac{XY}{f} \\ -\frac{f}{z} & \frac{Y}{z} & f + \frac{Y^2}{f} \end{pmatrix} \qquad (3)$$

where z represents the depth of each projected point P in the image plane, and f is the camera focal (see figure 1(b)).

B. Motivations and hypotheses

Now, we focus on the problem of estimating image data whenever they are lost during the vision-based task. Thus, the key-assumption of this work is that the whole image is considered to be **momentarily totally unavailable**. Therefore, methods dedicated to partial occlusions such as [9][10] cannot be used here. It is also possible to consider other approaches based on signal processing techniques or tracking methods such as [11][12]... However, most of this kind of techniques rely on measures from the image which is considered to be totally unavailable in our case. Moreover, recalling that our estimation method will be used in the control law design, it must provide the estimated visual signals sufficiently rapidly with respect to the control sampling period. Therefore, we need a simple and fast method. We have then thought of using a 3D model of the landmark together with projective geometry. However, this method requires to determine accurately the calibration parameters to reconstruct successfully the lacking data. Thus, we propose another approach consisting in solving (2) using the previous visual measurements and the control inputs $\dot{\mathbf{q}}$. Therefore, we have made the two following hypotheses: first, we assume that the image features can be measured at the beginning of the robotic task. Second, we suppose that few errors occur on the model and on the measures (we will see in section IV that this hypothesis is reliable).

C. Problem Statement

Our idea is to solve the dynamic system (2) to determine the image data $\boldsymbol{s}$. However, this last system depends on depth $\boldsymbol{z}$ which must then be determined. As our robot is not equipped with any sensor able to measure this data (see section IV), we have to reconstruct it. To this aim, different approaches may be used [13][14][15][16]. However, most of these techniques rely on measures from the image and cannot be applied here. In this paper, we propose another solution consisting in simultaneously estimating depth $\boldsymbol{z}$ and image data $\boldsymbol{s}$. It is then necessary to express the analytical relation between the variation of $\boldsymbol{z}$ and the camera motion. As the considered landmark is made of n points, it can be easily shown that, for one 3D point p projected into a point $P(X, Y)$ in the image plane as shown in figure 1(b), the depth variation $\dot{z}$ is related to the camera motion according to: $\dot{z} = \mathbf{L}_{(\boldsymbol{z})}\mathbf{J}_{\mathrm{r}}\dot{\mathbf{q}}$, where $\mathbf{L}_{(\boldsymbol{z})} = (0,\ -1,\ \frac{z}{f}Y)$. Thus, the dynamic system to be solved for one point $P(X, Y)$ expresses as follows:

$$\begin{cases} \dot{X} = \frac{X}{z} V_{\vec{\mathbf{z}}_C} + \frac{XY}{f} \Omega_{\vec{\mathbf{x}}_C} \\ \dot{Y} = -\frac{f}{z} V_{\vec{\mathbf{y}}_C} + \frac{Y}{z} V_{\vec{\mathbf{z}}_C} + \left(f + \frac{Y^2}{f}\right) \Omega_{\vec{\mathbf{x}}_C} \\ \dot{z} \ = -V_{\vec{\mathbf{z}}_C} - \frac{z}{f} Y \Omega_{\vec{\mathbf{x}}_C} \end{cases} \qquad (4)$$

Now, we address the resolution problem. A first idea is to integrate the above system for any $t \in [t_0, t_f]$ where t_0 and t_f are respectively the initial and final instants of the task. However, the computations are quite difficult to carry out. Therefore, in this work, we propose to discretize the problem and to solve system (4) during a time control interval $[t_k, t_{k+1}]$. Defining $\psi = [X_1, Y_1, \ldots, X_n, Y_n, z_1, \ldots, z_n]^T$ and $\mathbf{L}_{(\psi)}= \left[\mathbf{L}_{(s)}{}^T,\, \mathbf{L}_{(\boldsymbol{z})}{}^T\right]^T$, the differential system to be solved for n points is given by:

$$\begin{cases} \dot{\psi}(t) \ \ = \mathbf{L}_{(\psi_k)}\, \mathbf{J}_{\mathrm{r}}(\vartheta(t))\, \dot{\mathbf{q}}(t_k) \\ \psi(t_k) = \psi_k = \left[\boldsymbol{s}_k^T,\, \boldsymbol{z}_k^T\right]^T \end{cases}, \qquad \forall t \in [t_k, t_{k+1}] \quad (5)$$

where $\dot{\mathbf{q}}(t_k)= (v_k,\, \omega_k,\, \varpi_k)^T$ is the control input at t_k. Note that, on our robot, the control input is hold during the control law sampling period T_{s}. We will then consider in the sequel that $\dot{\mathbf{q}}(t_k)$ remains constant during this time interval.

D. Our estimation method

A first idea is to solve differential system (5) using numerical schemes as done in [17][18]. The main advantage of such approaches lies in their generality. Indeed, they can be applied to any set of visual features and to any kind of robots, provided that the estimation problem can be expressed as system (5). However, with such methods, the accuracy of the obtained solution depends on the chosen algorithm and on the integration step size. This is the reason why we propose in this work to solve analytically system (5). In this way, we obtain its **exact** solution on $[t_k, t_{k+1}]$ and the accuracy of the result is significantly improved. After some computations (see [19] for a detailed proof), it can be shown that, for any $t \in [t_k, t_{k+1}]$, the evolution of abscissa X and ordinate Y of one point P in the image expresses as:

$$\begin{cases} X(t) = \frac{z_k X_k}{z(t)} \\ Y(t) = f\, \frac{\dot{z} + v_k \cos(\vartheta(t)) + D_x \omega_k \sin(\vartheta(t)) - C_y(\omega_k + \varpi_k)}{(\omega_k + \varpi_k) z(t)} \\ z(t) \ = c_1 \sin\left(A_1\left(t - t_k\right)\right) + c_2 \cos\left(A_1\left(t - t_k\right)\right) \\ \qquad -D_x \cos\left(\vartheta(t)\right) + \frac{v_k}{\omega_k} \sin\left(\vartheta(t)\right) - C_x \end{cases} \qquad (6)$$

where:

$$\begin{cases} A_1 = (\omega_k + \varpi_k) \\ c_1 \ = -\frac{Y_k z_k}{f} + D_x \sin\left(\vartheta_k\right) + \frac{v_k}{\omega_k} \cos\left(\vartheta_k\right) - C_y \\ c_2 \ = z_k + D_x \cos\left(\vartheta_k\right) - \frac{v_k}{\omega_k} \sin\left(\vartheta_k\right) + C_x \end{cases}$$

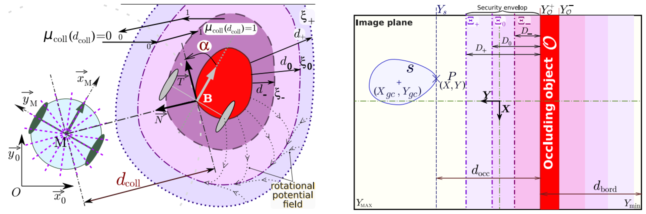
As one can see, the solution (6) require the determination of $\vartheta(t)$. This angle can simply be computed by integrating $\dot{\vartheta} = \varpi$ between t_k and t. Some straightforward calculus leads to $\vartheta(t) = \varpi_k\left(t - t_k\right) + \vartheta_k$, where ϑ_k is the pan-platform angular value at t_k, which is usually provided by the embedded encoder.

The solution for the set of the n points of the landmark is then easily obtained by applying the above solution (6) on each component of the vector ψ.

III. Control strategy

Let us recall that our goal is to perform vision-based navigation tasks in environments cluttered with occluding obstacles, which requires to deal with occlusions and collisions. The first problem will be treated thanks to our estimation method, and the second one using a potential field technique. So, first of all, we will briefly describe how to detect the risks of occlusion and collision. Then, we will detail our control strategy, considering two cases when the occlusion occur in the free space or in the obstacle neighborhood. Finally, we will end this part by some simulation results.

A. *Collision and Occlusion Detection*



(a) Obstacle detection (from ultrasonic sensors).

(b) Occlusion detection (provided by image processing).

Fig. 2. Avoidance modeling.

In this part, our goal is to detect the risks of collision and occlusion. To this aim, we design two parameters μ_{coll} and μ_{occ} which smoothly increase between 0 (no risk) and 1 (high risk). From the embedded ultrasonic sensors we can compute the shortest distance d_{coll} and the relative orientation α between the robot and the obstacle (see figure 2(a)). Defining by ξ_+, ξ_0, ξ_- three envelopes surrounding each obstacle at distance $d_+ > d_0 > d_-$, it is possible to choose $\mu_{\text{coll}}(d_{\text{coll}})$ so that it smoothly increases from 0 when the robot is far from the obstacle ($d_{\text{coll}} > d_0$) to 1 when it is close to it ($d_{\text{coll}} < d_-$). Assuming that the image processing is able to identify any occluding object $\mathcal{O}$, μ_{occ} can be determined by detecting their left and right borders. Indeed, from these data, we can deduce the shortest distance d_{occ} between $\boldsymbol{s}$ and $\mathcal{O}$ (see figure 2(b)). Introducing three envelopes Ξ_+, Ξ_0, Ξ_- around the occluding object located at $D_+ > D_0 > D_-$ from it, $\mu_{\text{occ}}(d_{\text{occ}})$ can be chosen to smoothly increase from 0 when $\mathcal{O}$ is far from the visual features ($d_{\text{occ}} > D_0$) to 1 when it is close to them ($d_{\text{occ}} < D_-$). Further details about these modelling, and different μ_{coll} and μ_{occ} expression can be found in [6] and [7].

B. *Control in the free space*

Our goal is to position the embedded camera with respect to a landmark made of n points despite possible occlusions. To this aim, we have applied the visual servoing technique given in [8] to mobile robots as in [20]. The proposed approach relies on the task function formalism [21] and consists in expressing the visual servoing task by the following task function to be regulated to zero:

$$e_{\text{vs}} = \mathbf{L}_{(\boldsymbol{s}^\star)}{}^+(\boldsymbol{s} - \boldsymbol{s}^\star) \tag{7}$$

where $\boldsymbol{s}^\star$ represents the desired value of the image data. Classically, a controller making e_{vs} vanish can be designed by imposing an exponential decrease, that is: $\dot{e}_{\text{vs}} = -\lambda_{\text{vs}} e_{\text{vs}}$, where λ_{vs} is a positive scalar or a positive definite matrix. Then, the visual servoing controller can be written as follows:

$$\dot{\mathbf{q}}_{(\boldsymbol{s})} = \mathbf{J}_{\text{r}}^{-1}(-\lambda_{\text{vs}})\mathbf{L}_{(\boldsymbol{s}^\star)}{}^+(\boldsymbol{s} - \boldsymbol{s}^\star) \tag{8}$$

Now, we suppose that an obstacle enters the camera field of view without inducing a danger of occlusion. In such a case, this controller cannot be computed anymore, leading to a task failure. Here, our idea is to embed our estimation technique in control law (8) to treat successfully this unexpected event. We simply propose to replace in expression (8) the real visual features $\boldsymbol{s}$ by their values $\widetilde{s}$ computed by our method. In case of occlusions, the visual servoing controller is then given by: $\dot{\mathbf{q}}_{(\widetilde{\boldsymbol{s}})} = \mathbf{J}_{\text{r}}^{-1}(-\lambda_{\text{vs}})\mathbf{L}_{(\boldsymbol{s}^\star)}{}^+(\widetilde{\boldsymbol{s}} - \boldsymbol{s}^\star)$. Now, it remains to switch between $\dot{\mathbf{q}}_{(\boldsymbol{s})}$ and $\dot{\mathbf{q}}_{(\widetilde{\boldsymbol{s}})}$ depending on the landmark visibility. We propose the following final visual servoing controller:

$$\dot{\mathbf{q}}_{\text{VS}} = (1 - \mu_{\text{occ}})\dot{\mathbf{q}}_{(\boldsymbol{s})} + \mu_{\text{occ}}\dot{\mathbf{q}}_{(\widetilde{\boldsymbol{s}})} \tag{9}$$

***Remark** 1:* There is no need to smooth controller (9) when the image features are lost. Indeed, when the occlusion occurs, as the last provided data are used to feed our reconstruction algorithm, the values of $\boldsymbol{s}$ and $\widetilde{\boldsymbol{s}}$ are close and so are $\dot{\mathbf{q}}_{(\boldsymbol{s})}$ and $\dot{\mathbf{q}}_{(\widetilde{\boldsymbol{s}})}$. Usually, the same reasoning holds when the occluding object leaves the image. However, some smoothing may be useful if the camera motion has been unexpectedly perturbed during the estimation phase or if the algorithm has been given too inaccurate initial conditions.

C. *Control in the obstacle neighborhood*

Now, we address the problem of controlling the robot in the obstacle neighborhood. This is the most complex part of our work because, during this phase, both risks of collision and occlusion may occur simultaneously (or not!). Then, we propose to design a multi-sensor-based controller able to guarantee the robot safety and to treat total image loss if needed. To this aim, we propose to extend the controller elaborated in [22] which only allows to avoid obstacles while tracking the landmark. We first recall the chosen control strategy and then explain how to embed our estimation technique to deal with the occlusions.

1) Obstacle avoidance strategy: The chosen strategy consists in defining 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 2(a)). 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 propose the following potential function $\mathcal{U}_{(d_{\text{coll}})}$ [22]:

$$\begin{cases} \mathcal{U}_{(d_{\text{coll}})} = \frac{1}{2}k_1\left(\frac{1}{d_{\text{coll}}} - \frac{1}{d_+}\right)^2 + \frac{1}{2}k_2\left(d_{\text{coll}} - d_+\right)^2 & \text{if } d_{\text{coll}} \leq d_+ \\ \mathcal{U}_{(d_{\text{coll}})} = 0 & \text{otherwise} \end{cases}$$

where k_1 and k_2 are positive gains to be chosen. The virtual repulsive force is defined to generate the desired rotative potential field around the obstacle by the couple (F, β), 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 wrt. $\mathcal{F}_M$. The mobile base velocities v_{coll} and ω_{coll} are then given by [22]:

$$\dot{\mathbf{q}}_{\text{mb}} = \begin{pmatrix} v_{\text{coll}} & \omega_{\text{coll}} \end{pmatrix}^T = \begin{pmatrix} k_v F \cos\beta & \frac{k_\omega}{D_x} F \sin\beta \end{pmatrix}^T \tag{10}$$

where k_v and k_ω are positive gains to be chosen. $\dot{\mathbf{q}}_{\text{mb}}$ 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 vision-based task at the end of the avoidance phase. Therefore, our idea is to control the pan-platform to keep the landmark in the camera line of sight during the avoidance motion. To this aim, as the camera is constrained to move within an horizontal plane, it is sufficient to regulate to zero the error $e_{\mathrm{gc}} = Y_{\mathrm{gc}} - Y_{\mathrm{gc}}^{\star}$ where Y_{gc} and $Y_{\mathrm{gc}}^{\star}$ are the current and desired ordinates of the target gravity center. Rewriting equation (1) as: $\mathbf{v}_{\mathrm{r}}^{c} = \mathbf{J}_{\mathrm{mb}}\dot{\mathbf{q}}_{\mathrm{mb}} + \mathbf{J}_{\varpi}\varpi_{\mathrm{coll}}$, and imposing an exponential decrease to regulate e_{gc} to zero ($\dot{e}_{\mathrm{gc}} = -\lambda_{\mathrm{gc}} e_{\mathrm{gc}}$, $\lambda_{\mathrm{gc}} > 0$), we finally obtain [22]:

$$\varpi_{\mathrm{coll}} = \frac{-1}{\mathbf{L}_{(Y_{\mathrm{gc}})}\mathbf{J}_{\varpi}}(\lambda_{\mathrm{gc}} e_{\mathrm{gc}} + \mathbf{L}_{(Y_{\mathrm{gc}})}\mathbf{J}_{\mathrm{mb}}\dot{\mathbf{q}}_{\mathrm{mb}}) \tag{11}$$

where $\mathbf{L}_{(Y_{\mathrm{gc}})}$ is the 2nd row of $\mathbf{L}_{(P)}$ evaluated for Y_{gc}.

2) *Extension to deal with occlusions during avoidance:*

In the obstacle vicinity, the visual data may enter and leave the image plane depending on the relative robot-obstacle configuration and on the obstacle height. We may even imagine that the camera could be occluded by another obstacle than the one which is currently avoided. We need then to design a control law able to deal with these sudden and unexpected modifications. Our idea is then to use our estimation method to provide the necessary visual data to the pan-platform control law whenever needed. Thus, when an occlusion occurs, our algorithm is fed with the last available measures and computes an estimation $\widetilde{\boldsymbol{s}}$ of the "real" visual features $\boldsymbol{s}$ from which we can deduce $\widetilde{Y_{\mathrm{gc}}}$. The pan-platform controller during an occlusion phase will then be determined by replacing Y_{gc} by its estimate $\widetilde{Y_{\mathrm{gc}}}$ in (11). We get:

$$\widetilde{\varpi}_{\mathrm{coll}} = \frac{-1}{\mathbf{L}_{(\widetilde{Y_{\mathrm{gc}}})}\mathbf{J}_{\varpi}}(\lambda_{\mathrm{gc}} \widetilde{e}_{\mathrm{gc}} + \mathbf{L}_{(\widetilde{Y_{\mathrm{gc}}})}\mathbf{J}_{\mathrm{mb}}\dot{\mathbf{q}}_{\mathrm{mb}}), \tag{12}$$

where $\widetilde{e}_{\mathrm{gc}} = \widetilde{Y_{\mathrm{gc}}} - Y_{\mathrm{gc}}^{\star}$, and $\mathbf{L}_{(\widetilde{Y_{\mathrm{gc}}})}$ is deduced from (3). Now, it remains to select the suitable controller depending on the availability of the image. Recalling that μ_{occ} allows to detect occlusions, we propose the following avoidance controller:

$$\dot{\mathbf{q}}_{\mathrm{coll}} = \left(v_{\mathrm{coll}}, \quad \omega_{\mathrm{coll}}, \quad (1-\mu_{\mathrm{occ}})\varpi_{\mathrm{coll}} + \mu_{\mathrm{occ}}\widetilde{\varpi}_{\mathrm{coll}}\right)^{T} \tag{13}$$

Thanks to this controller, it is possible to avoid the obstacle while treating the image total loss (if needed).

D. *Global control strategy*

So, we have previously designed two controllers given by (9) and (13). The first one allows to execute a vision-based task despite occlusions; the second one guarantees obstacle avoidance while treating image loss if necessary. To execute our vision-based navigation task, we have to apply to the robot the right controller depending on the risks of occlusions and collisions. Several approaches allow to sequence tasks [20][23][24]. Here, we have chosen a method relying on convex combinations between the successive controllers [22], as it allows to carry out applications more easily. We propose the following global control law:

$$\dot{\mathbf{q}} = (1-\mu_{\mathrm{coll}})\,\dot{\mathbf{q}}_{\mathrm{VS}} + \mu_{\mathrm{coll}}\dot{\mathbf{q}}_{\mathrm{coll}} \tag{14}$$

where $\dot{\mathbf{q}}_{\mathrm{VS}}$ and $\dot{\mathbf{q}}_{\mathrm{coll}}$= $(v_{\mathrm{coll}}, \omega_{\mathrm{coll}}, \varpi_{\mathrm{coll}})^{T}$ are respectively given by equations (9) and (13).

Remark 2: μ_{coll} and μ_{occ} are defined to be maintained to 1 once they have reached this value [6]. Moreover, the different envelops are chosen close enough to reduce the transition phase duration. Thus the control strategy is built to insure that the robot will be rapidly controlled by the most relevant controller. In this way, the risks of instability, target loss or collisions during the switch are significantly reduced and the task feasibility can be considered to be guaranteed.

E. *Simulation results*

To validate our work, we have simulated a mission whose objective is to position the camera relatively to a landmark made of $n = 9$ points. Moreover, the environment has been cluttered with two obstacles which may occlude the camera or represent a danger for the mobile base. Let us notice that the sampling period has been defined to be close to our real robot one, that is $T_{\mathrm{s}} = 50$ms.

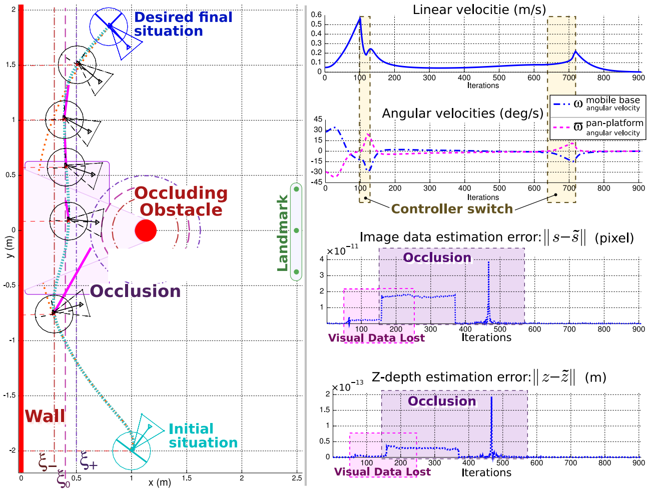


Fig. 3. Simulation results.

As shown on figure 3, the vision-based navigation task is correctly performed despite the presence of occluding obstacles. At the beginning of the task, there is no risk of collision. The robot is only controlled by $\dot{\mathbf{q}}_{\mathrm{VS}} = \dot{\mathbf{q}}_{(\boldsymbol{s})}$ and starts converging towards the target. After 50 steps, we simulate an image processing failure (for example due to a sudden illumination change). Our estimation procedure is then enabled and the robot is driven using $\dot{\mathbf{q}}_{\mathrm{VS}} = \dot{\mathbf{q}}_{(\widetilde{\boldsymbol{s}})}$ during 200 steps. Meanwhile, the vehicle enters the vicinity of the wall. Then, the global controller progressively switches to $\dot{\mathbf{q}}_{\mathrm{coll}}$ and the robot is constrained to follow the security envelope ξ_0 while tracking a *virtual* target until the image data is available anew. This event occurs when the circular obstacle leaves the camera field of view, allowing the pan-platform control to smoothly switches from $\widetilde{\varpi}_{\mathrm{coll}}$ to ϖ_{coll}. When there is no more danger, the control switches back to $\dot{\mathbf{q}}_{\mathrm{VS}} = \dot{\mathbf{q}}_{(\boldsymbol{s})}$ and the robot perfectly realizes the desired task. Figure 3 shows also the values of the estimation errors norms $\|\boldsymbol{s} - \widetilde{\boldsymbol{s}}\|$ and $\|\boldsymbol{z} - \widetilde{\boldsymbol{z}}\|$ which both remain close to zero. This result appears to be consistent, as our estimation method is based on a model which reflects precisely the behaviour of the system.

IV. EXPERIMENTAL ASPECTS

A. *The robotic platform*

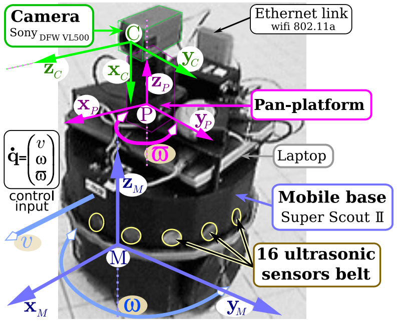


Fig. 4. The Nomadic Super Scout II.

We have implemented the above control law on the mobile robot Super Scout II[1] (see figure 4). This is a cylindric cart-like vehicle dedicated to indoor navigation. Incremental encoders provide the approximate velocity of the robotic system. The robot is equipped with 16 ultrasonic sensors from which we compute the shortest distance d_{coll} and the relative orientation α between the robot and the obstacle. A DFW-VL500 Sony color digital IEEE1394 camera captures pictures in YUV4 : 2 : 2 format with 640 × 480 resolution. The robot is controlled by an on-board laptop computer running under Linux on which is installed a specific control architecture called G$^{\text{en}}$oM (Generator of Module) [25].

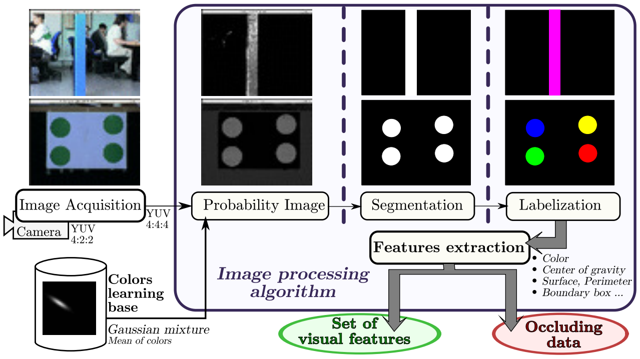
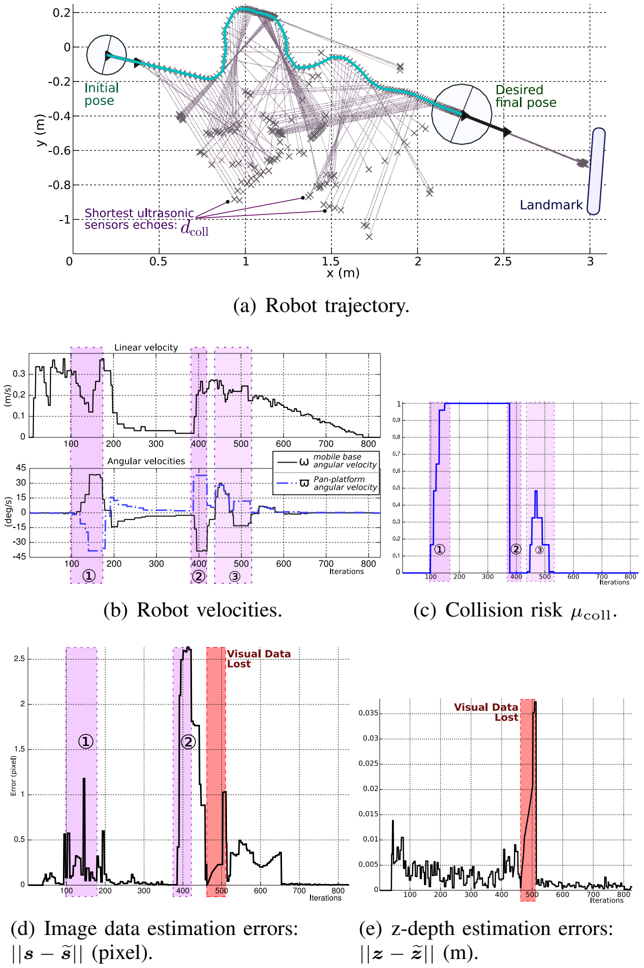


Fig. 5. Visual features and occluding object extraction algorithm pipeline.

The implementation of our control strategy requires different basic services. They are provided by two already existing modules respectively dedicated to the robotic system control and to data sensory processing. First, we have designed a specific module to manage the camera features. It allows to initialize and configure the camera, acquire pictures and process them to extract the visual data $\boldsymbol{s}$ and the occluding object position from which we compute d_{occ}. The image processing pipeline is shown on figure 5. It relies on the following hypotheses to reduce the computation time: all the occluding objects share the same color, the occluding object and the target colors are different, and finally, an occluding object detected in the image can never be positioned *behind* the target in the 3D scene. We have then embedded our control strategy in a second module which offers the required services to realize vision-based navigation tasks amidst obstacles: robotic system initialization, software emergency stop, parameters settings (choice of the different control gains, of the desired visual features s^*...), execution of a vision-based task with or without taking into account the occlusion and/or collision phenomena. We present below the obtained experimentation results.

B. *Experimental Results*

We consider a vision-based navigation task which consists in positioning the embedded camera in front of a given landmark made of $n = 4$ points. The environment is cluttered with one obstacle, and envelopes ξ_-, ξ_0 and ξ_+ are respectively located at $d_- = 40$cm, $d_0 = 56$cm and $d_+ = 70$cm from the obstacle. The obtained results are presented on figure 6.



(a) Robot trajectory.

(b) Robot velocities.

(c) Collision risk μ_{coll}.

(d) Image data estimation errors: $||\boldsymbol{s} - \tilde{\boldsymbol{s}}||$ (pixel).

(e) z-depth estimation errors: $||\boldsymbol{z} - \tilde{\boldsymbol{z}}||$ (m).

Fig. 6. Visual servoing with obstacle avoidance, experimental results.

Figure 6(a) shows the trajectory performed by the robot. At the beginning of the task, there is no risk of collision, and the image data are available. The robot is controlled using $\dot{\mathbf{q}}_{\text{VS}}$ and starts converging towards the landmark. It

[1] The Super Scout II is provided by the AIP-PRIMECA.

enters the vicinity of the obstacle, inducing a danger of collision. μ_{coll} increases to reach 1 (see figure 6(c)) and the control law applied to the vehicle smoothly switches to controller $\dot{q}_{\text{coll}}$ (13). Thus, the linear velocity decreases while the angular velocities ω_{coll} and ϖ_{coll} vary to make the robot avoid the obstacle and track the target (see figure 6(b)). When the obstacle is overcome, μ_{coll} vanishes, and the control progressively switches back to $\dot{\mathbf{q}}_{\text{VS}} = \dot{\mathbf{q}}_{(\boldsymbol{s})}$. However, at the same time, the landmark is manually occluded and the robot is driven using controller $\dot{\mathbf{q}}_{(\widetilde{\boldsymbol{s}})}$ instead of $\dot{\mathbf{q}}_{(\boldsymbol{s})}$, keeping on converging towards the target until $\boldsymbol{s}$ reaches $\boldsymbol{s}^*$. Thus, thanks to our method, it is possible to complete a task which would have been aborted otherwise.

Finally, we compare the obtained simulated and experimental results. To this aim, we analyse the quality of the image features reconstruction which appears to be better in simulation as one could have expected. To exhibit the parameters which influence the estimation error, we first recall that our algorithm is initialized using the last visual data available before their lost and on the base of the robot velocities. Therefore, the estimation errors depends mainly on the feature extraction noise (here about 1 pixel) and on the accuracy of the real robot velocities measurement. As the velocities executed by the robot are provided by the embedded encoders with a small delay and drift, we can observe an increase of the estimation error when the control input $\dot{\mathbf{q}}$ switches between the successive controllers (see figures 6(b) and 6(d)). Furthermore, as our platform runs under Linux which is not a real-time operating system, the sampling period T_{s} is not known accurately. These two aspects reduces the efficiency of reconstruction when validating on our real robot. However, despite these problems, figures 6(d) and 6(e) show that the estimation errors remain small, demonstrating the validity of our approach in an experimental context.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have elaborated a multi-sensor-based control law allowing a robot to perform safely a vision-based task in an unknown environment despite occlusions. We have first proposed a technique able to reconstruct the visual features when they are suddenly lost. Then, we have integrated it our control law to realize the desired mission. The method relies on the switch between different controllers depending on the risks of collision and occlusion. We have finally validated our work by experimental results.

However, the developed analytical solutions are restricted to landmarks which can be characterized by points and dedicated to a particular robotic system. Thus a natural extension of these works would be to solve this problem for other kinds of visual features and more complex robotic platform. We have also considered a perfect modelling. Therefore, it would be also interesting to evaluate signal processing approaches such as Kalman filtering to take into account the effects of noise on the reconstruction. Finally, a dynamical sequence of the controllers could also be interesting to provide better theoretic feasibility conditions.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[2] Y. Mezouar and F. Chaumette, "Avoiding self-occlusions and preserving visibility by path planning in the image," *Robotics and Autonomous Systems*, vol. 41, no. 2, p. 77–87, Nov. 2002.

[3] 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, Jul. 2003.

[4] V. Kyrki, D. Kragic, and H. Christensen, "New shortest-path approaches to visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct. 2004, pp. 349–355.

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

[6] 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 Conf.*, Seville, Spain, Dec. 2005.

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

[8] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, pp. 313–326, Jun. 1992.

[9] A. I. Comport, E. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, Oct. 2004.

[10] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Trans. Robot. Autom.*, vol. 21, pp. 1214–1220, 2005.

[11] P. Favaro and S. Soatto, "Seeing beyond occlusions (and other marvels of a finite lens aperture)," in *IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, Saint Louis, USA, 2003.

[12] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects," *Found. Trends. Comput. Graph. Vis*, vol. 1, no. 1, 2006.

[13] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth in image sequences," *Int. Journal of Computer Vision*, vol. 3, no. 3, pp. 209–238, 1989.

[14] C. Jerian and R. Jain, "Structure from motion: A critical analysis of methods," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 572–588, May 1991.

[15] S. Thrun, D. Fox, W. Burgard, and F. Dallaert, "Robust mote-carlo localization for mobile robots," *Artifial Intelligence*, vol. 128, no. 1-2, pp. 99–141, May 2001.

[16] J. Oliensis, "Exact two-image structure from controlled motion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1618–1633, Dec. 2002.

[17] D. Folio and V. Cadenat, "A new controller to perform safe vision-based navigation tasks amidst possibly occluding obstacles," in *European Control Conf.*, Kos, Greece, Jul. 2007.

[18] ——, "Using simple numerical schemes to compute visual features whenever unavailable," in *Int. Conf. on Informatics in Control Automation and Robotics*, Angers, France, May 2007.

[19] 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, University of Toulouse, France, Jul. 2007.

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

[21] C. Samson, M. L. Borgne, and B. Espiau, *Robot Control : The task function approach*. Oxford science publications, 1991.

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

[23] P. Souères and V. Cadenat, "Dynamical sequence of multi-sensor based tasks for mobile robots navigation," in *Symposium on Robot Control*, Wroclaw, Poland, Sep. 2003.

[24] N. Mansard and F. Chaumette, "Tasks sequencing for visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, Sep. 2004.

[25] S. Fleury and M. Herrb, *$G^{en}oM$: User Manual*, LAAS-CNRS, 2001. [Online]. Available: http://www.laas.fr/~sara/RIA/RIA-genom.html