

A new controller to perform safe vision-based navigation tasks amidst possibly occluding obstacles

David FOLIO and Viviane CADENAT

Abstract—This paper provides a new method allowing to safely perform a vision-based task amidst possibly occluding obstacles. Our objective is to fulfill the following two requirements: first, the robot safety by guaranteeing noncollision; second, the ability of keeping on realizing the vision-based task despite possible target occlusions or loss. To this aim, several controllers have been designed and then merged depending on the risks of collisions and occlusions. The presented simulation results validate the proposed approach.

I. INTRODUCTION

Visual servoing techniques aim at controlling the robot motion using visual features provided by a camera mounted on the robot or fixed to the environment [1][2]. The approaches that allow to design such control laws require that the image features remain always visible [3][4][5] and cannot be used anymore as soon as the target is lost or occluded. Thus the visual features visibility during a vision-based task appears as an interesting and challenging problem. Classically, the proposed solutions aim at *avoiding* occlusions and loss. Different solutions are proposed in the literature. For example, visibility may be preserved by path-planning in the image [6], by acting on specific DOFs [7][8][9] or benefiting from redundancy [10][11], by controlling the zoom [12] or, more recently, by making a tradeoff with the nominal vision-based task [13].

However, most of these works are dedicated to manipulator arms. 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 [14][15]. The developed methods allow to avoid collisions, occlusions and target loss when executing a vision-based task amidst obstacles. However, these methods are restricted to navigation tasks for which the simultaneous avoidance of both occlusions and obstacles is possible. Therefore, a true extension of these works would be to provide methods which accept that occlusions *may effectively occur* during the avoidance phase. A first solution to this problem is to allow *some* of the features to appear and disappear temporarily from the image as done in [16]. However, this approach is limited to partial occlusions.

In this paper, our goal is to realize a positioning vision-based task amidst possibly occluding obstacles. We aim at extending the works [14][15] by designing control laws able

to **accept** that occlusions may effectively occur when bypassing obstacles. Our control strategy relies on the definition of three different controllers: the first one realizes the nominal vision-based task in the free space, the second one guarantees occlusion avoidance when there is no risk of collision, while the third one drives the robot in the obstacle neighborhood. This last controller will have to guarantee non collision and to insure that the task execution continue even if the visual features are temporary unavailable. The global control law is obtained by switching between these controllers depending on the environment.

The article is organized as follows. Section II is devoted to preliminaries and problem statement. The different controllers are described in the next two sections. The global strategy and the simulation results are given in section V.

II. PRELIMINARIES

A. Modelling

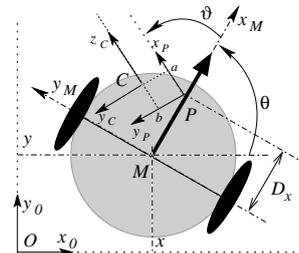


Fig. 1. The mobile robot with pan-platform

We consider the model of a cart-like robot with a camera mounted on a pan-platform as shown on figure 1. We define the successive frames: $\mathcal{F}_M (M, x_M, y_M, z_M)$ linked to the robot, $\mathcal{F}_P (P, x_P, y_P, z_P)$ attached to the pan-platform, and $\mathcal{F}_C (C, x_C, y_C, z_C)$ linked to the camera. Let (x, y) be the coordinates of the robot reference point M with respect to the world frame \mathcal{F}_O . θ and ϑ are respectively the directions of the vehicle and of the pan-platform with respect to the x_O and x_M axes. P is the pan-platform center of rotation, D_x the distance between M and P . The transformation between \mathcal{F}_P and \mathcal{F}_C is deduced from a hand-eye calibration method. It consists of an horizontal translation of vector $(a, b, 0)^T$ and a rotation of angle $\frac{\pi}{2}$ about the y_P -axis. 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 . Let $T^c = (V^c_{\mathcal{F}_C/\mathcal{F}_O}, \Omega^c_{\mathcal{F}_C/\mathcal{F}_O})^T$ be the kinematic screw representing the translational and rotational velocity

D. Folio is a PhD. thesis student supported by the European Social Fund. The authors are with the LAAS-CNRS, Toulouse, France, and the Paul Sabatier University, Toulouse, France; {dfolio, cadenat}@laas.fr

of \mathcal{F}_C with respect to \mathcal{F}_O , expressed in the frame \mathcal{F}_C . The kinematic screw is related to the joint velocity vector by the robot jacobian J , that is: $T^c = J \dot{q}$. As the camera is constrained to move horizontally, it is sufficient to consider a reduced kinematic screw $T_r^c = (V_{y_c}, V_{z_c}, \Omega_{x_c})^T$, and a reduced jacobian matrix J_r as follows:

$$T_r^c = \begin{pmatrix} -\sin(\vartheta) & D_x \cos(\vartheta) + a & a \\ \cos(\vartheta) & D_x \sin(\vartheta) - b & -b \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} \quad (1)$$

B. Obstacle detection

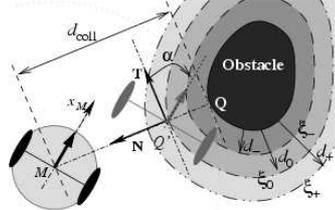


Fig. 2. Obstacle avoidance

The robot is equipped with proximetric sensors which provide a set of data characterizing locally the closest obstacle, that is any obstacle located at a distance less than d_+ (see figure 2). We then process these data to obtain a couple (d_{coll}, α) , where d_{coll} is the signed distance between M and the closest point Q on the obstacle, and α is the angle between the tangent to the obstacle at Q and the robot direction. The chosen avoidance strategy requires to characterize the *risk of collision*. To this aim, we define three envelopes around each obstacle (see figure 2). The first one ξ_+ located at a distance $d_+ > 0$ surrounds the zone inside which the obstacle is detected by the robot. The second one ξ_0 , located at a distance $d_0 > 0$ constitutes the virtual path along which the vehicle will move to pass around the obstacle. The last one ξ_- defines the region inside which the risk of collision is maximal. Now, from these definitions, we are able to characterize the *risk of collision* by the following parameter μ_{coll} which is defined to smoothly increase from 0 when the robot is far from the obstacle ($d_{coll} > d_0$) to 1 when it is close to it ($d_{coll} < d_-$). μ_{coll} is given by:

$$\begin{cases} \mu_{coll} = 0 & \text{if } d_{coll} > d_0 \text{ and AVOID}=0 \text{ and LEAVE}=0 \\ \mu_{coll} = \frac{d_{coll} - d_0}{d_- - d_0} & \text{if } d_{coll} \in [d_-, d_0] \text{ and AVOID}=0 \\ \mu_{coll} = \frac{d_{coll} - d_+}{d_L - d_+} & \text{if } d_{coll} \in [d_L, d_+] \text{ and LEAVE}=1 \\ \mu_{coll} = 1 & \text{otherwise} \end{cases}, \quad (2)$$

where AVOID is a flag enabled when the robot enters the zone surrounded by ξ_- , and LEAVE is set to 1 when the robot is allowed to leave the obstacle neighborhood. d_L is equal to the distance d_{coll} when LEAVE = 1. The use of μ_{coll} in the global control strategy will be detailed in section V.

C. Occlusion detection

Now, we address the problem of detecting occlusions. We suppose that an occluding object \mathcal{O} lies in the camera line of sight. Its projection appears in the image plane as

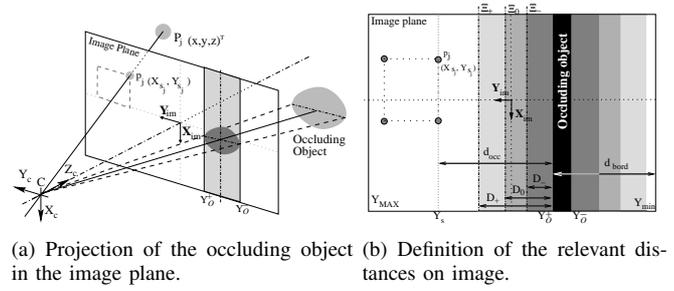


Fig. 3.

shown on figure 3(a) and we denote by Y_O^- and Y_O^+ the ordinates of its left and right borders. As the camera is constrained to move in the horizontal plane, there is no loss of generality in stating the occlusion detection on Y_O^- and Y_O^+ . Considering figure 3(b), we denote by (X_{s_j}, Y_{s_j}) the coordinates of each point P_j of the target in the image frame, Y_{min} and Y_{max} representing the ordinates of the two image sides. It is then possible to define the distance before occlusion by the shortest distance between the image features and the occluding object as follows:

$$d_{occ} = \min \left(\min_j |Y_{s_j} - Y_O^+|, \min_j |Y_{s_j} - Y_O^-| \right) = |Y_s - Y_{occ}|,$$

where Y_s is the ordinate of the closest point P_j to object \mathcal{O} , while Y_{occ} represents the closest border of \mathcal{O} to the visual features (in the case of figure 3(b), $Y_{occ} = Y_O^+$).

The occlusion avoidance strategy requires to characterize the *risk of occlusion*. We then introduce the following distances:

- d_{bord} is defined by the distance separating the occluding object \mathcal{O} and the opposite image side to the visual features:

$$d_{bord} = \min (|Y_O^+ - Y_{max}|, |Y_O^- - Y_{min}|) = |Y_{occ} - Y_{bord}|$$

- D_+ defines an envelope Ξ_+ delimiting the region inside which the risk of occlusion is detected.

- D_0 and D_- define two additional envelopes Ξ_0 and Ξ_- . They respectively surround the critical zone inside which it is necessary to start avoiding occlusion and the region where the danger of occlusion is the highest. The *risk of occlusion* can then be modelled by the following μ_{occ} which smoothly increases from 0 when \mathcal{O} is far from the visual features ($d_{occ} > D_0$) to 1 when it is close to them ($d_{occ} < D_-$):

$$\begin{cases} \mu_{occ} = 0 & \text{if } d_{occ} > D_0 \text{ and OCCLU}=0 \\ \mu_{occ} = \frac{d_{occ} - D_0}{D_- - D_0} & \text{if } d_{occ} \in [D_-, D_0] \text{ and OCCLU}=0 \\ \mu_{occ} = \frac{d_{occ} - D_+}{D_L - D_+} & \text{if } d_{occ} \in [D_L, D_+] \\ \mu_{occ} = 1 & \text{otherwise} \end{cases}, \quad (3)$$

where OCCLU is a flag which is set to 1 when the occluding object enters the zone defined by Ξ_- . D_L is defined by the distance d_{occ} when OCCLU is disabled. The use of μ_{occ} in the global control strategy will be detailed in section V.

D. Problem statement and preliminary assumptions

1) *Problem statement*: We consider the problem of determining a sensor-based closed-loop controller for driving

the robot until the camera is positioned in front of a target amidst possibly occluding obstacles. Two objectives have to be fulfilled: first, the robot motion must be collision free; second, the ability of keeping on realizing the vision-based task despite possible target occlusions or loss must be guaranteed. We have splitted the problem into two parts concerning respectively the control law design when the robot is far and close to the obstacles.

2) *Hypotheses*: For the problem to be well stated, we assume that no obstacle nor occluding object lie in a close goal neighborhood. We also suppose that the distance between two obstacles is greater than $2d_+$ to prevent the robot from considering several obstacles simultaneously. Finally, we assume that any object which has been recognized as an occluding object by the image processing algorithm lies between the camera and the target in the 3D scene (see remark 2).

III. CONTROL DESIGN WHEN FAR FROM THE OBSTACLES

In this part, we consider that the robot is far from the obstacles, that is there is no risk of collision. Therefore, two cases may occur: either there is no risk of occlusion and a classical visual servoing control can be used; or, an occluding object enters the camera field of view and a specific control law dedicated to occlusion avoidance must be designed. These two different cases are analyzed in the following subsections.

Remark 1: The presence of an occluding object in the image does not necessarily mean that a collision may occur. Indeed, an obstacle may be detected by the camera before it becomes dangerous for the mobile base. This is the reason why we consider two different controllers depending on the occlusion occurs far from the obstacle or close to it.

A. The nominal visual servoing control

First, we present the nominal vision-based controller in the case that occlusions and collisions do not occur. We consider the visual servoing technique introduced in [3]. This approach relies on the task function formalism, which consists in expressing the desired task as a task function e to be regulated to zero [17]. A sufficient condition that guarantees the control problem to be well conditioned is that e is ρ -admissible. Indeed, this property ensures the existence of a diffeomorphism between the task space and the state space, so that the ideal trajectory $q_r(t)$ corresponding to $e = 0$ is unique. This condition is fulfilled if $\frac{\partial e}{\partial q}$ is regular around q_r [17].

In our application, the target is made of N points, defining a $2N$ -dimensional vector of visual signals s in the camera plane. At each configuration of the robot, the variation of the signals \dot{s} is related to the kinematic screw \mathcal{T}_r^c by the interaction matrix \mathcal{L} [3]:

$$\dot{s} = \mathcal{L} \mathcal{T}_r^c \quad (4)$$

For a point p of coordinates $(x, y, z)^T$ in \mathcal{F}_C projected into a point $P(X, Y)$ in the image plane (see figure 3(a)), \mathcal{L} is directly deduced from the optic flow equations [3] and

given by the following matrix which has a reduced number of columns to be compatible with the dimension of \mathcal{T}_r^c :

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_X \\ \mathcal{L}_Y \end{bmatrix} = \begin{bmatrix} 0 & \frac{X}{z} & XY \\ -\frac{1}{z} & \frac{Y}{z} & 1 + Y^2 \end{bmatrix} \quad (5)$$

Following the task function formalism, the task is defined as the regulation of an error function $e_{vs}(q(t))$ to zero:

$$e_{vs}(q(t)) = C(s(q(t)) - s^*), \quad (6)$$

where s^* is the desired value of the visual signal and $q = [l, \theta, \vartheta]^T$, l representing the curvilinear abscissa of the robot. $C = (\mathcal{L}^T \mathcal{L})^{-1} \mathcal{L}^T$ is a full-rank $32N$ combination matrix¹ which allows to take into account more visual features than available degrees of freedom [3]. Classically, a kinematic controller can be determined by imposing an exponential convergence of e_{vs} to zero: $\dot{e}_{vs} = C \mathcal{L} \mathbf{J}_r \dot{q} = \mathbf{J}_r \dot{q} = -\lambda_{vs} e_{vs}$, where λ_{vs} is a positive scalar or a positive definite matrix. From this last relation together with (1), (4) and (6), we get²:

$$\dot{q}_{vs} = \mathbf{J}_r^{-1}(-\lambda_{vs})e_{vs} = \mathbf{J}_r^{-1}(-\lambda_{vs})\mathcal{C}(s(q, t) - s^*) \quad (7)$$

B. Occlusion avoidance

Now, we suppose that an occluding object \mathcal{O} lies in the camera line of sight as in figure 3(a). Our objective is to *avoid* occlusions. Indeed, as the robot is considered to be far from the obstacle, such a choice is not too restrictive as the main limitations come from the simultaneous avoidance of both occlusions and collisions [14][15]. We propose to use the same approach as in [14]. In this work, we have used the redundant task function formalism [17] to preserve the visual features visibility. We have chosen to define the occlusion avoidance as the priority redundant task, $e_{\mathcal{O}}$, and the secondary goal to keep on tracking the target. Modelling the latter by a cost function h_s to be minimized under the constraint that the primary task $e_{\mathcal{O}}$ is perfectly performed (that is, $e_{\mathcal{O}} = 0$), the redundant task function formalism shows that this optimization problem is equivalent to the regulation to zero of the following task function e_{occ} [17]:

$$e_{occ}(q(t)) = W_{\mathcal{O}}^+ e_{\mathcal{O}} + \beta_{occ}(I - W_{\mathcal{O}}^+ W_{\mathcal{O}})g_s, \quad (8)$$

where $W_{\mathcal{O}} = \frac{\partial e_{\mathcal{O}}}{\partial q}$, β_{occ} is a positive scalar, $g_s = \frac{\partial h_s}{\partial q}$. Our strategy consists in preventing the visual features from being occluded (that is, $d_{occ} \neq 0$) while making the occluding object go out of the image ($d_{bord} = 0$). A redundant task allowing to fulfill these two objectives is given by (see [14] for details):

$$e_{\mathcal{O}} = \left(\tan\left(\frac{\pi}{2} - \frac{\pi}{2} \cdot \frac{d_{occ}}{D_+}\right) \quad d_{bord} \right)^T$$

The secondary task is defined to track the target and is modelled by the following criterion h_s :

$$h_s = \frac{1}{2}(s - s^*)^T (s - s^*) \Rightarrow g_s = ((s - s^*)^T \mathcal{L} \mathbf{J}_r)^T$$

¹See [3] and [17] for more details about the choice of \mathcal{C} .

²Note that e_{vs} is a ρ -admissible task as \mathbf{J}_r is always invertible ($\det(\mathbf{J}_r) = -D_x$). No singularity occurs in the determination of \dot{q}_{vs} .

As $W_{\mathcal{O}}$ and β_{occ} are chosen to fulfill the assumptions of the redundant task formalism [17], the task jacobian $\frac{\partial e_{\text{occ}}}{\partial q}$ is positive definite around the ideal trajectory and e_{occ} is ρ -admissible. This result also allows to simplify the control synthesis as it can be shown that a controller making e_{occ} vanish is given by [3]:

$$\dot{q}_{\text{occ}} = -\lambda_{\text{occ}} e_{\text{occ}}, \quad (9)$$

where λ_{occ} is a positive scalar or a positive definite matrix.

IV. CONTROL DESIGN WHEN CLOSE TO THE OBSTACLES

Now, we consider that the robot is close to an obstacle. Therefore, there is a collision risk, but an occlusion may also occur, depending on the obstacle height. In such a case, a strategy which aims at simultaneously avoiding both occlusions and collisions appears to be limited because it is restricted to missions where this avoidance motion is possible [15][14]. This is why we have designed a new strategy allowing to *accept* that the image features may be lost during the collision avoidance motion. The idea consists in controlling separately the mobile base and the pan-platform: the first one will be driven to avoid the obstacle and the second one to track the “true” target if the visual signals are available or a “predicted” one if an occlusion occurs. These controllers are designed in the next subsections.

A. Mobile base control law design

The literature provides many techniques to navigate safely in a cluttered environment. Here, we have chosen to make the robot follow the security envelope ξ_0 to pass round the obstacle as shown on figure 2. To this aim, we have used the path following formalism introduced in [18]. We define a mobile frame on ξ_0 whose origin Q' is the orthogonal projection of M (see figure 2). During obstacle avoidance, the robot linear velocity is supposed to be kept constant. Let $\delta = d_{\text{coll}} - d_0$ be the signed distance between M and Q' . With respect to the moving frame, the dynamics of the error terms (δ, α) is described by the following system:

$$\begin{cases} \dot{\delta} = v \sin \alpha \\ \dot{\alpha} = \omega - v \chi \cos \alpha \end{cases} \quad \text{with } \chi = \frac{\frac{\sigma}{R}}{1 + \frac{\sigma}{R} \cdot \delta} \quad (10)$$

where $\sigma = \{-1, 0, +1\}$ depending on the sense of the robot motion around the obstacle and R is the curvature radius of the obstacle. The path following problem is classically defined as the search for a controller ω allowing to steer the pair (δ, α) to $(0, 0)$ under the assumption that v never vanishes to preserve the system controllability. To guarantee this property, we propose to fix v to a nonzero constant value v_{coll} chosen small enough to let the robot sufficiently slow down to avoid collision when entering the critical zone. The literature provides numerous different path following control laws. We have chosen to use the one proposed in [19]:

$$\omega_{\text{coll}} = -v_{\text{coll}}(k\delta + \alpha + 2k \sin \alpha - \chi \cos \alpha), \quad (11)$$

where $k > 0$ is a gain to be fixed. A complete stability proof based on Lyapunov theory and specific values of k guaranteeing noncollision and actuators nonsaturation are

provided in [19]. The mobile base avoidance controller is then defined by:

$$\dot{q}_{\text{base}} = \begin{pmatrix} v_{\text{coll}} & \omega_{\text{coll}} \end{pmatrix}^T \quad (12)$$

B. Pan-platform control law design

During the avoidance phase, if the pan-platform is not controlled, e_{vs} will increase, preventing from switching back to the nominal vision-based task, once the obstacle is overcome. Our idea is to control the pan-platform to compensate the mobile base motion while tracking the “true” target if the visual features remain available during the avoidance phase or a “virtual” one if not. We first address the problem of image features computation before dealing with the pan-platform control design.

1) *Visual features computation:* To compute the image features when necessary, we have chosen to numerically integrate (4). However, as \mathcal{L} depends on depth, it is also necessary to evaluate this parameter together with the visual data s . It can be shown that, for a 3D point $p_i(x_i, y_i, z_i)^T$ in \mathcal{F}_C projected into a point $P_i(X_i, Y_i)$ in the image plane, the point velocity $V_{p_i} = -V_{C/\mathcal{F}_O} - \Omega_{\mathcal{F}_C/\mathcal{F}_O} \wedge p_i$. The depth variation \dot{z}_i is then related to the camera kinematic screw by: $\dot{z}_i = \mathcal{L}_{z_i} T_r^c$, where $\mathcal{L}_{z_i} = \begin{bmatrix} -1 & z_i Y_i & z_i X_i \end{bmatrix}$. From this last relation and equation (4) and denoting by $\psi = (X_1, Y_1, z_1, \dots, X_4, Y_4, z_4)^T$, the differential equations to be solved are given by:

$$\dot{\psi} = \begin{pmatrix} \mathcal{L}_1 \\ \vdots \\ \mathcal{L}_4 \end{pmatrix} T_r^c = \mathbf{F}(\psi), \quad (13)$$

where $\mathcal{L}_i = (\mathcal{L}_{X_i}^T, \mathcal{L}_{Y_i}^T, \mathcal{L}_{z_i}^T)^T$, \mathcal{L}_{X_i} and \mathcal{L}_{Y_i} being computed from (5). To numerically integrate (13), we assume that the image features previous values and the camera kinematic screw are known. The first hypothesis is not restrictive as it suffices to launch our algorithm before the avoidance motion beginning; the second one is a bit more limitative as T_r^c is not completely determined independently from the computed visual signals \tilde{s} : indeed, only \dot{q}_{base} is defined by the avoidance motion as shown by (12), while ϖ_{coll} will depend on \tilde{s} . The pan-platform is then controlled with an open-loop scheme during the occlusion. Therefore, the proposed method is restricted to short duration occlusions, which is often the case when avoiding an obstacle. Note also that the open-loop control only affects the pan-platform, while the mobile base remains driven by a closed-loop scheme. Noncollision is then still preserved.

The literature provides many methods allowing to numerically solve ordinary differential equations (ODE) [20]. We have tested several schemes to solve (13). We have finally chosen to use the Backward Differentiation Formulas (BDF), also known as Gear’s formulas [21]. Indeed, the BDF involves few ODE evaluations, which is an advantage for real-time applications. Moreover, it has been shown to be particularly well suitable to solve stiff problems, that is problems where the integration step size h is more severely limited by the stability of the method than its accuracy

[20]. This kind of problem occurs when the differential system (13) varies very rapidly, which is our case during the avoidance phase. The BDF approach solves the ODE (13) by interpolating $\psi_k, \psi_{k-1}, \dots, \psi_{k-m}$ using a polynomial $g(t)$ of degree $m + 1$. Denoting by h the integration step, the following relation presents the fourth order BDF scheme that we have used for our purpose [21]:

$$\psi_{k+1} = \frac{48}{25}\psi_k - \frac{36}{25}\psi_{k-1} + \frac{16}{25}\psi_{k-2} - \frac{3}{25}\psi_{k-3} + \frac{12}{25}h\mathbf{F}(\tilde{\psi}_{k+1}, t_{k+1}) \quad (14)$$

Note that the BDF is an implicit scheme, that is it requires a first computation $\tilde{\psi}_{k+1}$ of ψ_{k+1} . This first approximation can be obtained by a simple Euler method or any other explicit scheme.

Remark 2: Recalling that our method allows to reconstruct both the visual features and their depths, we are currently working on extending this approach to compute the depth of the occluding object. In this way, it will be possible to know whether an object which appears in the camera field of view really induces a risk of occlusion or not. The previously mentioned third assumption will then be useless.

2) *Pan-platform control design:* Now, let us address the ϖ_{coll} design problem. When considering available image features, the proposed approach is similar to [19]. The idea is to control the pan-platform 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 $e_{\text{gc}} = Y_{\text{gc}} - Y_{\text{gc}}^*$ where Y_{gc} and Y_{gc}^* are the current and desired ordinates of the target gravity center. Separating the terms relative to the mobile base velocities from the ones relative to the pan-platform velocity in J_r , (1) can be rewritten as follows: $T_r^c = [J_{\text{base}} J_{\varpi}] \dot{q} = J_{\text{base}} \dot{q}_{\text{base}} + J_{\varpi} \varpi_{\text{coll}}$. Now, imposing an exponential decrease to regulate e_{gc} to zero, we get:

$$\dot{e}_{\text{gc}} = \mathcal{L}_{Y_{\text{gc}}} T_r^c = \mathcal{L}_{Y_{\text{gc}}} J_{\text{base}} \dot{q}_{\text{base}} + \mathcal{L}_{Y_{\text{gc}}} J_{\varpi} \varpi_{\text{coll}} = -\lambda_{\text{gc}} e_{\text{gc}}, \quad (15)$$

where $\mathcal{L}_{Y_{\text{gc}}}$ is the 2nd row of \mathcal{L} evaluated for Y_{gc} and λ_{gc} is a positive scalar or a positive definite matrix. From this relation, ϖ_{coll} is then given by³:

$$\varpi_{\text{coll}} = \frac{-1}{\mathcal{L}_{Y_{\text{gc}}} J_{\varpi}} (\lambda_{\text{gc}} e_{\text{gc}} + \mathcal{L}_{Y_{\text{gc}}} J_{\text{base}} \dot{q}_{\text{base}}) \quad (16)$$

However, the previous equation allows to determine ϖ_{coll} only if Y_{gc} can be deduced from s . When an occlusion occurs, s is no more available and the pan-platform cannot be controlled anymore using (16). At this time, we compute the visual features by integrating the ODE (13) using the fourth order BDF (14). It is then possible to keep on executing the previous task e_{gc} , even if the visual features are temporary occluded by an obstacle. The pan-platform controller during an occlusion phase will then be deduced by replacing the real target gravity center ordinate Y_{gc} by the computed one \tilde{Y}_{gc} in (16). We get:

$$\tilde{\varpi}_{\text{coll}} = \frac{-1}{\tilde{\mathcal{L}}_{Y_{\text{gc}}} J_{\varpi}} (\lambda_{\text{gc}} \tilde{e}_{\text{gc}} + \tilde{\mathcal{L}}_{Y_{\text{gc}}} J_{\text{base}} \dot{q}_{\text{base}}), \quad (17)$$

³In our case, it can be proven $\mathcal{L}_{Y_{\text{gc}}} J_{\varpi} \neq 0$, preserving the ρ -admissibility of e_{gc} and the existence of ϖ_{coll} .

where $\tilde{e}_{\text{gc}} = \tilde{Y}_{\text{gc}} - Y_{\text{gc}}^*$ and $\tilde{\mathcal{L}}_{Y_{\text{gc}}}$ is deduced from (5). Now, it remains to apply the suitable controller to the pan-platform depending on the context. Recalling that the parameter $\mu_{\text{occ}} \in [0; 1]$ has been designed to detect occlusions, we propose the following controller to drive the robot in the obstacle vicinity:

$$\dot{q}_{\text{coll}} = (v_{\text{coll}} \quad \omega_{\text{coll}} \quad (1 - \mu_{\text{occ}})\varpi_{\text{coll}} + \mu_{\text{occ}}\tilde{\varpi}_{\text{coll}})^T, \quad (18)$$

where v_{coll} and ω_{coll} are defined by (12), while ϖ_{coll} and $\tilde{\varpi}_{\text{coll}}$ are given by (16) and (17).

V. APPLICATION

We first present one navigation strategy based on the previously designed controllers, before describing the simulation results.

A. Global navigation strategy

Our global control strategy relies on the detection of the risks of collisions and occlusions and therefore on the two previously defined parameters μ_{coll} (2) and μ_{occ} (3). These parameters smoothly increases from 0 (no risk of collision, resp. occlusion), to 1 when these risks are maximum. Therefore, when $\mu_{\text{occ}} = \mu_{\text{coll}} = 0$, only \dot{q}_{vs} is applied to the robot. When μ_{occ} increases, if $\mu_{\text{coll}} = 0$, the vehicle is progressively controlled by \dot{q}_{occ} and starts avoiding the occlusion until the occluding object leaves the image ($d_{\text{bord}} = 0$) or goes out the critical zone ($d_{\text{occ}} \geq D_0$). When μ_{coll} increases and reaches 1, the risk of collision is maximal and the robot will be only driven by \dot{q}_{coll} . The mobile base and the pan-platform are then separately controlled to avoid the obstacle while tracking the target. If, during the avoidance phase, a risk of occlusion occurs, μ_{occ} increases, but \dot{q}_{coll} is always applied. As shown by (18), the mobile base remains driven to follow the security envelope, while the pan-platform is progressively controlled on the base of the computed visual features instead of the real ones. The robot safety is then preserved and the occlusion treated until μ_{occ} vanishes. The avoidance phase ends when the flag LEAVE is enabled, and, at this time, μ_{coll} is allowed to decrease to 0 (see (2)). The LEAVE condition is fulfilled when the obstacle is overcome (the directions of the mobile base and the pan-platform are identical: $\vartheta = 0$) if the target is not occluded ($\mu_{\text{occ}} = 0$). Otherwise (if $\mu_{\text{occ}} > 0$), the flag LEAVE is maintained to 0 until there is no more occlusion risk. In this way, we benefit from the avoidance motion to make the occluding object leave the image. Another possible solution would be to switch back to the vision-based task despite the occlusion and to design the visual servoing controller using the computed image features. However, such a choice would lead to control the robot with an open-loop scheme, which is not a satisfying solution for us. Finally, the whole reasoning is summarized in table I, and recalling that \dot{q}_{vs} , \dot{q}_{occ} and \dot{q}_{coll} are given by (7), (9) and (18), we propose the following global controller:

$$\dot{q} = (1 - \mu_{\text{occ}})(1 - \mu_{\text{coll}})\dot{q}_{\text{vs}} + (1 - \mu_{\text{coll}})\mu_{\text{occ}}\dot{q}_{\text{occ}} + \mu_{\text{coll}}\dot{q}_{\text{coll}} \quad (19)$$

	$\mu_{occ} = 0$	$\mu_{occ} \in]0, 1[$	$\mu_{occ} = 1$
$\mu_{coll} = 0$	$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{vs}$	$f(\dot{\mathbf{q}}_{vs}, \dot{\mathbf{q}}_{occ})$	$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{occ}$
$\mu_{coll} \in]0, 1[$	$f(\dot{\mathbf{q}}_{vs}, \dot{\mathbf{q}}_{coll})$	$f(\dot{\mathbf{q}}_{vs}, \dot{\mathbf{q}}_{occ}, \dot{\mathbf{q}}_{coll})$	$f(\dot{\mathbf{q}}_{occ}, \dot{\mathbf{q}}_{coll})$
$\mu_{coll} = 1$	$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{coll}$	$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{coll}$	$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{coll}$

TABLE I
THE SWITCHING STRATEGY.

Let us remark that the proposed control strategy relies on different sensory data: on one hand (d_{coll}, α) are computed from the proximetric sensors while on the other hand (d_{occ}, d_{bord}) are extracted from the image. Therefore, it clearly appears as a *multi-sensor-based* approach.

Remark 3: Note that μ_{occ} and μ_{coll} are maintained to 1 once they have reached this value and that the different envelops are chosen close enough to reduce the transition phase duration. 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, collisions or local minima during the switch are significantly reduced.

B. Simulation results

Our method has been implemented to simulate a mission whose objective is to position the camera in front of a given target made of $N = 25$ points. To validate our approach, the environment has been cluttered with two obstacles which may occlude the camera or represent a danger for the mobile base. The proposed method has been simulated using Matlab software. For this test, D_- , D_0 and D_+ have been respectively fixed to 40, 60 and 80 pixels, and d_- , d_0 , d_+ to 0.3m, 0.5m, and 0.6m. The sampling period and the integration step have been set to $T_e = h = 50$ ms. The obtained results are presented on figures 4 5, 6 and 7.

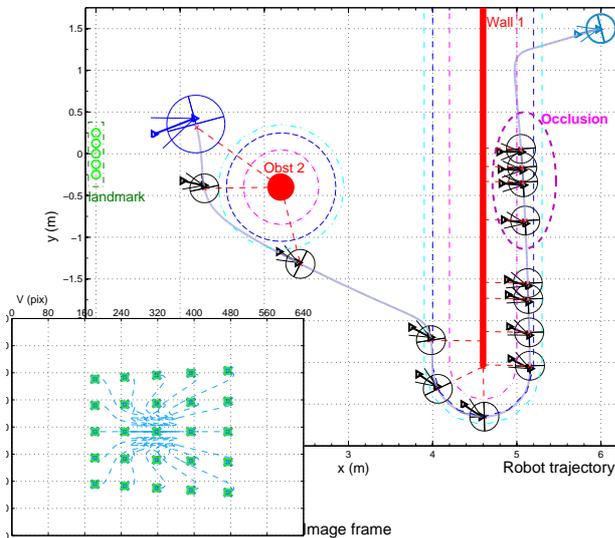


Fig. 4. Trajectories

As shown on figure 4, the vision-based navigation task is correctly performed. At the beginning of the task, as the first obstacle is lower than the camera, there is no risk of collision nor occlusion. The robot is only controlled by $\dot{\mathbf{q}}_{vs}$ and starts converging towards the target. When the vehicle enters the vicinity of the first encountered obstacle, μ_{coll} is

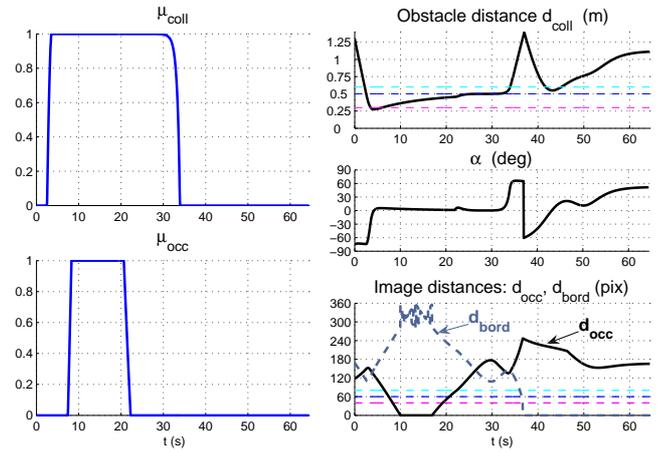


Fig. 5. Avoidance data

progressively increased to its maximum value (see figure 5), and the robot is constrained to follow the security envelope ξ_0 . During this phase, the second obstacle enters the camera field of view and μ_{occ} progressively increases to reach 1 when $d_{occ} < D_-$. At the same time, the BDF are used to compute the visual features and the pan-platform control smoothly switches from ϖ_{coll} to $\tilde{\varpi}_{coll}$. It is then possible to move around the obstacle while tracking a “virtual” target until the end of the occlusion. When the second obstacle leaves the image, μ_{occ} decreases and ϖ_{coll} is applied again to the pan-platform. As the maximal value of the euclidian norm of $\|s - \tilde{s}\|$ remains inferior to 0.17 pixel (see figure 6), the “real” target can be easily retrieved and tracked again while the mobile base keeps on following ξ_0 . The avoidance phase ends when the obstacle is overcome and $\mu_{occ} = 0$. At this time the flag LEAVE is enabled, μ_{coll} decreases and vanishes once ξ_+ is crossed. The sole controller $\dot{\mathbf{q}}_{vs}$ is applied to the vehicle and the camera finally reaches its desired position. The task is then successfully realized. Therefore, thanks to our method, the occlusion is only treated through the pan-platform control and does not affect the whole control law. In this way, local minima or deadlock situations which may arise when it becomes impossible to perform simultaneously collision and occlusion avoidance cannot occur. A wider range of navigation tasks are then realizable. Note that this navigation task could not have been executed with approaches [14][15] because of this last problem. The efficiency of our method is also demonstrated on figure 6 which shows that the euclidian norm of the errors $\|s - \tilde{s}\|$ and $\|z - \tilde{z}\|$ remain always inferior to respectively 0.17 pixel and 1.164mm for a 15s occlusion. Let us notice that when $\mu_{occ} = 0$, there is no need to evaluate the computed data, and the errors remain to 0 (see figure 6). Finally, figure 7 shows the obtained control inputs. They remain smooth despite the switches and they are consistent with the robot trajectory. Actuator saturation never occurs. The different steps of the navigation can be easily determined, especially the avoidance phase: it starts when ω increases to make the robot turn to bypass the obstacle while v is fixed to its security value 0.2m/s. The evolution of ϖ also shows the mobile base motion compensation during the avoidance.

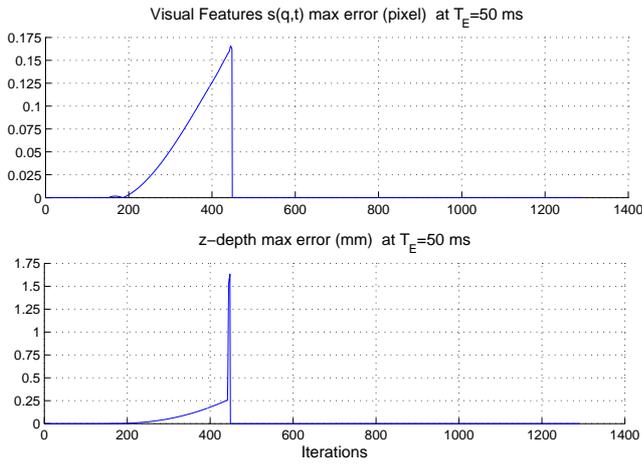


Fig. 6. Euclidian norms of $(s - \tilde{s})$ and $(z - \tilde{z})$

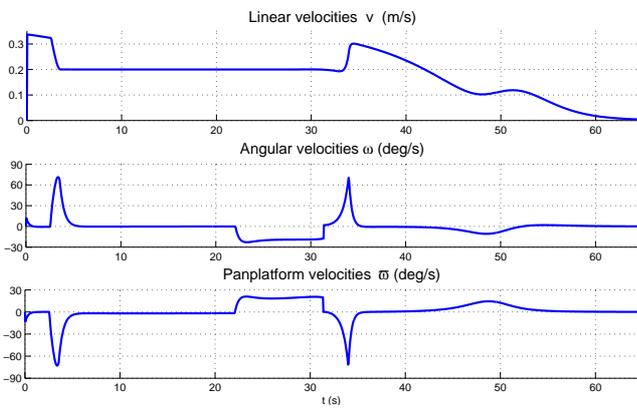


Fig. 7. Global controller velocities

VI. CONCLUSIONS AND FUTURE WORKS

The proposed control strategy allows to perform safely a vision-based task in a cluttered environment. Different sensor-based controllers have been designed and then merged depending on the environment to define the global control law. The main advantage of this approach is that **partial or total short duration occlusions** are now really **tolerated** in the obstacle neighborhood. It is then possible, contrary to previous works, to switch back to the nominal vision-based task at the end of the avoidance motion, even if the image features have been temporarily lost. By decoupling the mobile base and the pan-platform movement, noncollision can be guaranteed and local minima which arise when it is impossible to avoid both obstacles and occlusions suppressed. The range of realizable navigation tasks can then be enlarged.

However, the proposed method is limited to short duration occlusions which occur in the obstacles vicinity. A first extension of this work will consist in defining a controller able to drive the robot when the visual features remain unavailable at the end of the avoidance phase. Such an extension would prevent the robot from getting stuck around

the obstacle when the avoidance motion does not allow to make the occluding object leave the image. Furthermore, this approach may suffer from local minima due to the linear combination of several controllers. Although the control strategy has been chosen to reduce this kind of problem (see remark 3), it is impossible to strictly guarantee that it cannot occur. Therefore, we plan to modify the switching technique to suppress this problem by dynamically sequencing the different controllers [22]. Finally, we are currently addressing experimental issues to validate our control law on our mobile robot Super Scout II.

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," *IEEE Trans. Robot. Automat.*, Oct. 1996.
- [3] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, June 1992.
- [4] R. Pissard-Gibollet and P. Rives, "Applying visual servoing techniques to control a mobile hand-eye system," in *ICRA'95*, Nagoya, Japan, May 1995.
- [5] A. Crétual, "Asservissement visuel à partir d'information de mouvement dans l'image," Ph.D. dissertation, Université de Rennes 1, IRISA, Nov. 1998.
- [6] Y. Mezouar and F. Chaumette, "Avoiding self-occlusions and preserving visibility by path planning in the image," *Robotics and Autonomous Systems*, Nov. 2002.
- [7] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 507–515, Aug. 2001.
- [8] E. Malis, F. Chaumette, and S. Boudet, "2 1/2d visual servoing," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 238–250, Apr. 1999.
- [9] V. Kyrki, D. Kragic, and H. Christensen, "New shortest-path approaches to visual servoing," in *IROS'04*, Oct. 2004, pp. 349–355.
- [10] E. Marchand and G. Hager, "Dynamic sensor planning in visual servoing," in *ICRA'98*, Leuven, Belgium, May 1998.
- [11] N. Mansard and F. Chaumette, "A new redundancy formalism for avoidance in visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'05*, vol. 2, Edmonton, Canada, August 2005, pp. 1694–1700.
- [12] S. Benhimane and E. Malis, "Vision-based control with respect to planar and non-planar objects using a zooming camera," in *ICAR'03*, Coimbra, Portugal, July 2003.
- [13] A. Remazeilles, N. Mansard, and F. Chaumette, "Qualitative visual servoing: application to the visibility constraint," in *IROS'06*, Beijing, China, Oct. 2006.
- [14] D. Folio and V. Cadenat, "A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment," in *ECC-CDC'05*, Dec. 2005.
- [15] —, "Using redundancy to avoid simultaneously occlusions and collisions while performing a vision-based task amidst obstacles," in *ECMR'05*, Ancona, Italy, Sept. 2005.
- [16] 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. Automat.*, vol. 21, Dec. 2005.
- [17] C. Samson, M. L. Borgne, and B. Espiau, *Robot Control : The task function approach*, Clarendon Press ed. Oxford, England: Oxford science publications, 1991.
- [18] C. Samson, "Path following and time-varying feedback stabilization of a wheeled mobile robot," in *ICARCV'92*, Singapore, Sept. 1992.
- [19] V. Cadenat, P. Souères, and M. Courdresses, "An hybrid control for avoiding obstacles during a vision-based tracking task," in *ECC'99*, Karlsruhe, Germany, Sept. 1999.
- [20] L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations*. San Francisco: W. H. Freeman, 1975.
- [21] C. W. Gear, *Numerical initial value problems in ordinary differential equation*. Prentice-Hall, 1971.
- [22] P. Souères and V. Cadenat, "Dynamical sequence of multi-sensor based tasks for mobile robots navigation," in *SYROCO'03*, Wroclaw, Poland, Sept. 2003.