

A Redundancy-Based Approach for Visual Navigation with Collision Avoidance

Andrea Cherubini, F. Spindler, F. Chaumette

▶ **To cite this version:**

Andrea Cherubini, F. Spindler, F. Chaumette. A Redundancy-Based Approach for Visual Navigation with Collision Avoidance. IEEE Symp. on Computational Intelligence in Vehicles and Transportation Systems, CIVTS'11, 2011, Paris, France. hal-00639658

HAL Id: hal-00639658

https://inria.hal.science/hal-00639658

Submitted on 9 Nov 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Redundancy-Based Approach For Visual Navigation With Collision Avoidance

Andrea Cherubini, Fabien Spindler and François Chaumette

***Abstract*—We propose an autonomous vehicle guidance framework which combines visual navigation with simultaneous obstacle avoidance. The method was originally designed in [1], but real outdoor experiments and major improvements have been added in this paper. Kinematic redundancy guarantees that obstacle avoidance and navigation are achieved concurrently. The two tasks are realized both in an obstacle-free and in a dangerous context, and the control law is smoothened in between. The experiments show that with our method, the vehicle can replay a taught visual path while avoiding collisions.**

***Index Terms*—Navigation systems, Visual Servoing, Autonomous driving.**

I. INTRODUCTION

A great amount of robotics research focuses on vehicle guidance, with the ultimate goal of automatically reproducing the tasks usually performed by human drivers [2 – 6]. In many works, information from visual sensors is used for localization [7 – 9] or navigation [10]. In the case of autonomous navigation, an important task is obstacle avoidance, which consists of generating a collision-free trajectory to the goal [11]. If bypassing is impossible, the vehicle must decelerate to prevent collision [12]. A classical technique for avoidance is the potential field method [13], which is often associated to a global path planner.

The framework presented in our work does not require a global model of the environment (e.g., based on geometric primitives). It enables execution of a sensor-based task, with simultaneous collision avoidance. We focus on following a visual path, represented by a database of ordered key images [14]. The task is therefore defined in the image space, and not in the pose space, and it is divided into subtasks, each consisting of reaching the next key image in the database. This scheme, which is quite common in the robotics community [14 – 17], has been rarely extended to take into account obstacles.

Obstacle avoidance has been integrated in navigation frameworks in [18 – 22]. The authors of [18] use a model-based vision system with retroactive position correction. In [19], a laser range finder and monocular vision enable navigation in an office environment. Simultaneous obstacle avoidance and path following is presented in [20], where the geometry of the path (a curve on the ground) is perfectly known. The authors of [21] deform the desired trajectory to avoid sensed obstacles. In practice, all these methods are based on the environment 3D model, including, e.g. walls and doors. An exception is [22], where redundancy enables reactive obstacle avoidance, without requiring any 3D model.

Our scheme also exploits redundancy. A robot is redundant when it has more DOFs than those required for the primary task [23]; then, a secondary task can also be executed. Here, we focus on this problem: a wheeled vehicle, equipped with an actuated pinhole camera and with a forward-looking range scanner, must follow a visual path represented by key images, without colliding with the ground obstacles. The camera detects the features required for navigating, while the scanner senses the obstacles (in contrast with other works, such as [24], only one sensor is used to detect the obstacles). As in [22], we guarantee that obstacle avoidance has no effect on visual navigation. However, our controller is compact, and the transition between safe and dangerous contexts is operated only for the secondary task, while in [22], three controllers are needed, and the transitions are more complex.

Although the approach was designed in [1], the contributions of this paper are various. Here, we experimentally validate the scheme, and to our knowledge, this is the first time that obstacle avoidance and outdoor visual navigation are merged directly at the control level. Many changes were necessary in order to adapt the system in [1] to the real world: in particular, the linear velocity, which was constant in [1], had to be varied to improve visual tracking and security (it must diminish in dangerous situations). As a consequence, the task dimension and context definition also had to be modified. Under these real world constraints, we have designed a compact general controller, which provides visual navigation, obstacle circumnavigation, and collision avoidance by deceleration.

II. PROBLEM CHARACTERISTICS

The reader is referred to Fig. 1 for the definitions below. We define the robot frame $\mathcal{F}_{\mathcal{R}}\,(R, X', Y', Z')$ (R is the robot center of rotation), image frame $\mathcal{F}_{\mathcal{I}}(O, x, y)$ (O is the image center), and camera frame $\mathcal{F}_{\mathcal{C}}(C, X, Y, Z)$ (C is the optical center). The robot control inputs are:

$$\mathbf{u} = [v\ \omega\ \dot{\varphi}]^\top$$

These are, respectively, the linear and angular velocities of the vehicle, and the camera pan angular velocity. We use the normalized perspective camera model:

$$x = \frac{X}{Z}, \qquad y = \frac{Y}{Z}$$

We assume that $|\varphi| \leq \frac{\pi}{2}$, that C belongs to the camera pan rotation axis, and that the path can be tracked with continuous

A. Cherubini, F. Spindler and F. Chaumette are with INRIA Rennes - Bretagne Atlantique, IRISA, Campus de Beaulieu 35042, Rennes, France. {firstname.lastname}@inria.fr

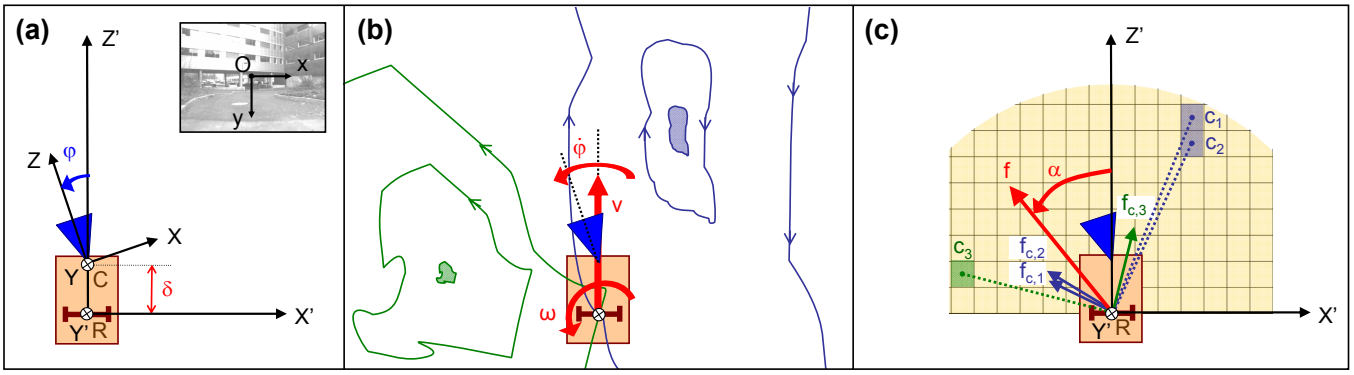


Fig. 1. Top view of the robot (orange), equipped with an actuated camera (blue). (a) Reference frames. (b) Obstacle-induced vortex fields, and control variables (v, ω, $\dot{\varphi}$). (c) Occupancy grid and potential field f construction.

$v\,(t) > 0$. This ensures safety, since only obstacles in front of the robot can be detected by our range scanner. The distance between R and C is denoted by $\delta \geq 0$.

The visual path that the robot must follow is represented as a database of ordered key images, such that each neighboring pair contains some common static visual features. First, the vehicle is manually driven along a *taught* path, with the camera pointing forward ($\varphi = 0$), and all the images are saved. Harris points [25] are detected in the first image, and a modified Kanade-Lucas-Tomasi (KLT) algorithm [26] tracks the features in successive images, to generate the database of N key images $I_1, \ldots, I_N$ (Fig. 2) that represents the path. Then, during autonomous navigation, the current image, noted I, is compared with the database, to drive the vehicle velocity. SIFT descriptors [27] are used to initialize the KLT algorithm, which tracks the features while the taught path is replayed. Visual navigation is divided into N subtasks, each consisting of driving the current image I to the next key image $\hat{I}$. To check whether $\hat{I}$ has been reached, we perform a coarse three-view pose calculation between the previous, current and next key images. As soon as $\hat{I}$ is reached, a topological transition is made, and the tracker is reinitialized: the next image in the set becomes the desired one, and so on, until I_N is reached. More details on these algorithms are in [14].

Along with the visual path following problem, which we have tackled in [14] and in [28], here we consider obstacles which are present on the path, but not in the database, and sensed by the range scanner. For obstacle modeling, as in [1], an occupancy grid, shown in Fig. 1(c), is used: it is linked to

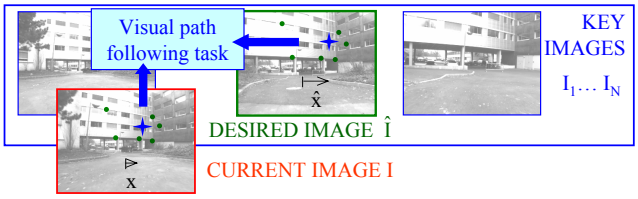


Fig. 2. The current and desired images contain some common visual features, which are used for navigation.

$\mathcal{F}_{\mathcal{R}}$, with cell sides parallel to the X' and Z' axes. Its forward and lateral extensions are smaller than the scanner radial range, to ignore obstacles that are too far away to jeopardize the robot. The grid is built from the latest $2K+1$ scans. For each cell centered at $c = [X'\ Z']^\top$, we define the $2K+1$ *occupancies* r at the j-th oldest iteration as:

$$r_j\,(c) = \{0, 1\}, \qquad j = 0, \ldots, 2K+1$$

We set $r_j = 1$ if an obstacle has been sensed in c at the j-th iteration prior to the current one, and 0 otherwise. Then, we associate to each cell a coefficient $\mu\,(c)$, obtained by linear combination of the occupancies, weighed with a normalized Gaussian filter that smoothens the cell effect over time:

$$\mu\,(c) = \sum_{j=0}^{2K+1} \frac{e^{-(j-K)^2/K}}{\sqrt{K\pi}}\ r_j\,(c)$$

The filter maximum weight is set at the K-th latest scan, to avoid the control inputs overshoot at a new obstacle detection (an issue which existed in [1]). If the robot velocity is negligible with respect to the scanner acquisition frequency, and K is reasonably small, the effect of motion on the occupancies can be ignored, and the above model is consistent with the current robot position. Later in this paper, we will explain how the occupancy grid is utilized in our controller, and we will show that the above assumptions are appropriate in our setup.

III. REDUNDANCY FRAMEWORK FOR 2 TASKS

We hereby recall the redundancy-based controller used in [1], and originally introduced in [29]. We name $\mathbf{s}_1 \in \mathbb{R}^k$ the primary task vector, and $\mathbf{u} \in \mathbb{R}^m$ the control inputs. Redundancy exists when $m > k$, and enables handling a secondary task, $\mathbf{s}_2 \in \mathbb{R}^{m-k}$. The tasks dynamics are related to the control inputs by:

$$\dot{\mathbf{s}}_i = \mathbf{J}_i \mathbf{u}, \qquad i = 1, 2 \tag{1}$$

where $\mathbf{J}_1$ and $\mathbf{J}_2$ are the *task jacobians*, respectively of size $k \times m$ and $(m-k) \times m$. We assume that both have full rank.

With the *Projected Gradient* method [23], the solution of (1), for tracking the primary task is:

$$\mathbf{u} = \mathbf{J}_1^{+} \dot{\mathbf{s}}_1^{*} + \mathbf{P}_1 \mathbf{h} \tag{2}$$

In the above equation:

- $\mathbf{J}_1^+ = \mathbf{J}_1^\top \left(\mathbf{J}_1 \mathbf{J}_1^\top\right)^{-1}$ is the $m \times k$ Moore-Penrose pseudoinverse of $\mathbf{J}_1$, i.e., a particular solution of: $\mathbf{J}_1 \mathbf{J}_1^+ = \mathbf{I}$;
- $\dot{\mathbf{s}}_1^* \in I\!R^k$ is the required primary task error evolution;
- $\mathbf{h} \in I\!R^m$ is an arbitrary control input;
- $\mathbf{P}_1 = \mathbf{I} - \mathbf{J}_1^+ \mathbf{J}_1$ projects $\mathbf{h}$ on the kernel of $\mathbf{J}_1$; hence, $\mathbf{P}_1 \mathbf{h}$ has no effect on the primary task.

Vector $\mathbf{h}$ can be used to apply a command that will carry out at best the secondary task $\mathbf{s}_2$, without disturbing $\mathbf{s}_1$.

Introducing (2) in (1) (with $i = 2$), we obtain:

$$\dot{\mathbf{s}}_2 = \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{s}}_1^* + \mathbf{J}_2 \mathbf{P}_1 \mathbf{h}$$

In [29], $\mathbf{h}$ is derived from this equation to track $\dot{\mathbf{s}}_2^*$:

$$\mathbf{h} = \left(\mathbf{J}_2 \mathbf{P}_1\right)^+ \left(\dot{\mathbf{s}}_2^* - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{s}}_1^*\right) \tag{3}$$

Plugging into (2), we obtain:

$$\mathbf{u} = \mathbf{J}_1^+ \dot{\mathbf{s}}_1^* + \mathbf{P}_1 \left(\mathbf{J}_2 \mathbf{P}_1\right)^+ \left(\dot{\mathbf{s}}_2^* - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{s}}_1^*\right)$$

The term $\mathbf{J}_2 \mathbf{P}_1$ gives the available range for $\mathbf{s}_2^*$ to be performed without affecting $\mathbf{s}_1$, while $\dot{\mathbf{s}}_2^* - \mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{s}}_1^*$ is the secondary task function, without the part $\mathbf{J}_2 \mathbf{J}_1^+ \dot{\mathbf{s}}_1^*$ accomplished by the first task. Since $\mathbf{P}_1$ is Hermitian and idempotent (it is a projection operator), the above equation becomes:

$$\mathbf{u} = \mathbf{J}_1^+ \dot{\mathbf{s}}_1^* + \mathbf{h} \tag{4}$$

with $\mathbf{h}$ defined in (3).

The desired evolution of the task errors can be written:

$$\dot{\mathbf{s}}_i^* = \dot{\hat{\mathbf{s}}}_i - \lambda_i \left(\mathbf{s}_i - \hat{\mathbf{s}}_i\right), \qquad i = 1, 2 \tag{5}$$

with $\hat{\mathbf{s}}_i$ and $\dot{\hat{\mathbf{s}}}_i$ indicating the desired values of the i-th task, and of its first derivative, and $\lambda_i > 0$ the associated gain.

Controller (4) has the following properties.

- It guarantees convergence of the primary task. Replacing (4) in (1) yields:

  $$\dot{\mathbf{s}}_1 = \dot{\mathbf{s}}_1^*$$

  Considering (5), this is equivalent to the linear system:

  $$\dot{\mathbf{s}}_1 - \dot{\hat{\mathbf{s}}}_1 = -\lambda_1 \left(\mathbf{s}_1 - \hat{\mathbf{s}}_1\right)$$

  for which, as desired, $\left(\hat{\mathbf{s}}_1, \dot{\hat{\mathbf{s}}}_1\right)$ is an exponentially stable equilibrium, since $\lambda_1 > 0$.
- The secondary task $\mathbf{s}_2$ has no effect on the primary task, since $\mathbf{h}$ is projected onto the null space of $\mathbf{J}_1$.
- The secondary task $\mathbf{s}_2$ is realized at best, under the constraint that it does not affect the primary task.

In this work, the desired specifications are:

1) orienting the camera to maximize the similarity of I with the next key image $\hat{I}$ in the database,
2) making the vehicle progress forward along the path,
3) avoiding collision with the obstacles, while remaining near the 3D taught path.

For our system, $m = 3$. We will fulfill a primary task of dimension $k = 1$ (orienting the camera), and use the 2 remaining degrees of freedom for a secondary task $\mathbf{s}_2 \in I\!R^2$ that realizes the two other specifications. We discern two contexts (*safe* and *dangerous*). To smoothen transitions, we design an *obstacle activation function*:

$$H : \mathcal{C} \mapsto [0, 1]$$

where the context $\mathcal{C}$ indicates the danger represented by detected obstacles. The definition of H will be given later.

- In the *safe context* ($H = 0$), since no obstacles are present, it is not necessary to deform the taught path. We use ω for the primary task (i.e., to orient the camera for image tracking). Then, we can set $\varphi = 0$, as was done during teaching. Moreover, the linear velocity v must be reduced in the presence of sharp turns, to ease the visual tracking of quickly moving features in the image. Hence, we formulate the secondary task as:

  $$\begin{cases} v = v_s \\ \dot{\varphi} = -\lambda_2 \varphi \end{cases} \qquad \forall \ (\mathbf{s}_1, \dot{\mathbf{s}}_1) \in I\!R \times I\!R \tag{6}$$

  The function v_s, which defines the linear velocity in the safe context, will be given later. Condition (6) will be guaranteed by control input $\mathbf{h}_s$, defined just below.
- In the *dangerous context* ($H > 0$), obstacles are present. Sometimes ($H < 1$), these can be circumnavigated, while tracking the visual features. In this case, the robot should reduce its velocity, and avoid collision by orienting its heading to a desired value α. The value of α is related to the obstacles position (see Fig. 1(c)), and defined later in the paper. The secondary task is:

  $$\begin{cases} v < v_s \\ \omega = \lambda_2 \alpha \end{cases} \qquad \forall \ (\mathbf{s}_1, \dot{\mathbf{s}}_1) \in I\!R \times I\!R \tag{7}$$

  This condition provokes a deformation of the taught 3D path. Correspondingly, the camera pan angle must be actuated to maintain visibility of the database features (i.e., the primary task). If the obstacles are too near to be circumnavigated ($H = 1$), the vehicle should stop:

  $$v = 0 \qquad \forall \ (\mathbf{s}_1, \dot{\mathbf{s}}_1) \in I\!R \times I\!R \tag{8}$$

  Conditions (7), and (8) will be guaranteed by control input $\mathbf{h}_d$, defined below.

A smooth transition between the inputs associated to the two contexts can be achieved by setting, in (4):

$$\mathbf{h} = H \mathbf{h}_d + (1 - H) \mathbf{h}_s$$

to obtain:

$$\begin{cases} \mathbf{u} = \mathbf{J}_1^+ \dot{\mathbf{s}}_1^* + H \mathbf{h}_d + (1 - H) \mathbf{h}_s \\ \mathbf{h}_d = \left(\mathbf{J}_d \mathbf{P}_1\right)^+ \left(\dot{\mathbf{s}}_d^* - \mathbf{J}_d \mathbf{J}_1^+ \dot{\mathbf{s}}_1^*\right) \\ \mathbf{h}_s = \left(\mathbf{J}_s \mathbf{P}_1\right)^+ \left(\dot{\mathbf{s}}_s^* - \mathbf{J}_s \mathbf{J}_1^+ \dot{\mathbf{s}}_1^*\right) \end{cases} \tag{9}$$

In the above equation:

- $\dot{\mathbf{s}}_{d,s}^*$ are the desired secondary tasks dynamics, in the dangerous and safe context; recalling (5), they are:

  $$\begin{cases} \dot{\mathbf{s}}_d^* = \dot{\hat{\mathbf{s}}}_d - \lambda_2 \left(\mathbf{s}_d - \hat{\mathbf{s}}_d\right) \\ \dot{\mathbf{s}}_s^* = \dot{\hat{\mathbf{s}}}_s - \lambda_2 \left(\mathbf{s}_s - \hat{\mathbf{s}}_s\right) \end{cases} \tag{10}$$

- $\mathbf{J}_{d,s}$ are the corresponding jacobian matrices.

In Sect. V, controller (9) will be instantiated for the problem of visual navigation with obstacle avoidance.

IV. SYSTEM MODELING

In this section, we define precisely the variables that have been introduced above. In particular, we will show how the linear velocity v_s for maximal visibility is designed, and how we use the occupancy grid to derive the robot heading for avoidance α, and the activation function H for varying between the safe and dangerous contexts.

A. Linear Velocity For Maximal Visibility

As we mentioned above, features can be lost in the presence of sharp turns. Hence, we relate the vehicle linear velocity v to the weighted difference between the abscissa of the centroids in the current image (x), and in the next ($\hat{x}$), and second next ($\hat{\hat{x}}$) key images[1]:

$$v_s = \frac{V}{2}\left[1 + \tanh\left(\pi - \frac{\left|2\hat{x} + \hat{\hat{x}} - 3x\right|}{\gamma}\right)\right] \qquad (11)$$

This function has an upper bound $V > 0$, and smoothly decreases to 0, as the abscissa difference grows. Its inflection point is determined by positive parameter γ.

B. Modeling The Obstacles

Obstacle avoidance is derived by using vortex potential fields [30]. For each cell c, we define the potential:

$$U_c = \frac{\mu(c)}{\|c\|}$$

where $\|c\|$ is the distance[2] of the cell from R. In practice, for two occupied cells with equal $\mu(c) \neq 0$, the nearest one will yield the highest potential. We define the vortex field for each cell simply as the rotor of U_c:

$$f_c = \begin{bmatrix} f_{c,X'} \\ f_{c,Z'} \end{bmatrix} = \begin{bmatrix} \pm\dfrac{\partial U}{\partial Z'} \\ \mp\dfrac{\partial U}{\partial X'} \end{bmatrix} = \mu(c) \begin{bmatrix} \mp\dfrac{Z'}{\|c\|^3} \\ \pm\dfrac{X'}{\|c\|^3} \end{bmatrix}$$

The signs of $f_{c,X'}$ and $f_{c,Z'}$ depend on the cell abscissa: positive (negative) X' will induce a clockwise (counterclockwise) vortex, so that the field always points forward. The fields $f_{c,i}$ generated by all the n cells c_i are then superimposed to obtain the total field:

$$f = \sum_{i=1}^{n} f_{c,i}$$

The orientation $\alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ of this field (see Fig. 1) is:

$$\alpha = \begin{cases} 0 & \text{if } f = 0 \\ -\text{ATAN2}\ (f_{X'}, f_{Z'}) & \text{otherwise} \end{cases}$$

As in [1] and [30], this is the desired heading for obstacle circumnavigation in dangerous situations (see (7)).

However, α is not indicative of the chances of successful circumnavigation. For this, we use the field norm:

$$|f| = \sqrt{f_{X'}^2 + f_{Z'}^2}$$

which is a good metric for evaluating the obstacle distance, and reducing v accordingly, in order to fulfill (7) and (8).

In summary, to assess the context danger, we consider both α and $|f|$, and using two empirically tuned thresholds ρ and ϱ such that $0 < \rho < \varrho$, we design the activation function as:

$$H = \begin{cases} \kappa\,|\alpha| & \text{if } |f| <= \rho \\ \kappa\,|\alpha| + (1 - \kappa\,|\alpha|)\,\frac{|f|-\rho}{\varrho-\rho} & \text{if } |f| \in]\rho, \varrho[\\ 1 & \text{if } |f| >= \varrho \end{cases}$$

Note that $H = 0$ if no obstacle is detected (since both $|f|$ and α are null in that case), and it is bounded by 1. For small $|f|$, H is determined only by α: the obstacles are 'far enough' to be circumnavigated using (7), and parameter $\kappa \in \left]0, \frac{2}{\pi}\right]$ weighs the danger provoked by the field orientation[3]. On the other hand, for large $|f| \geq \varrho$, the obstacles are 'too near': $H = 1$ and the robot stops. We use a linear function to guarantee continuity in between. Parameters ρ and ϱ are tuned to determine the three intervals. For example, small values of ϱ make the approach more conservative, since the robot will stop in the presence of farther obstacles.

V. CONTROL SCHEME

A. Primary Task: Camera Orientation

We do not take directly into account occlusions, or feature position, and assume that the visible features are homogeneously distributed in the environment. Then, the task of maximizing the similarity between current image I, and next key image $\hat{I}$, can be achieved with only one visual feature: the abscissa x of the centroid of the points matched on I and $\hat{I}$ [14]. Hence, for $\mathbf{s}_1$, we use:

$$\mathbf{s}_1 = x, \qquad \hat{\mathbf{s}}_1 = \hat{x}, \qquad \dot{\hat{\mathbf{s}}}_1 = 0$$

which, replaced in (5), yields:

$$\dot{\mathbf{s}}_1^* = \lambda_1(\hat{x} - x)$$

Assuming static features, the corresponding primary task Jacobian is:

$$\mathbf{J}_1 = [j_v\ j_\omega\ j_{\dot{\varphi}}], \qquad (12)$$

where an approximation of the components is (see [28]):

$$\begin{aligned} j_v &= \tfrac{-\sin\varphi + x\cos\varphi}{Z} \\ j_\omega &= \tfrac{\delta(\cos\varphi + x\sin\varphi)}{Z} + 1 + x^2 \\ j_{\dot{\varphi}} &= 1 + x^2 \end{aligned} \qquad (13)$$

As in [28], to avoid depth estimation, which can be unreliable, we set the depth to a fixed value Z.

[1] When the next image is the final one, we use $\hat{\hat{x}} = \hat{x}$, since the second next image is undefined.

[2] Designing the grid without the cell at $R = [0\ 0]^\top$ (where obstacles are not detectable by the range scanner), guarantees that U_c is non-singular.

[3] To guarantee $H \leq 1$, parameter κ must be chosen smaller than $\frac{2}{\pi}$.

The Moore-Penrose pseudoinverse of $\mathbf{J}_1$ is:

$$\mathbf{J}_1^+ = \frac{[j_v\ j_\omega\ j_{\dot\varphi}]^\top}{j_v^2 + j_\omega^2 + j_{\dot\varphi}^2}$$

Since $j_{\dot\varphi} \neq 0$, $\forall x \in I\!R$ (see (13)), $\mathbf{J}_1^+$ is full rank. The projecting operator $\mathbf{P}_1$ can also be derived:

$$\mathbf{P}_1 = \frac{1}{j_v^2 + j_\omega^2 + j_{\dot\varphi}^2} \begin{bmatrix} j_\omega^2 + j_{\dot\varphi}^2 & -j_v j_\omega & -j_v j_{\dot\varphi} \\ -j_v j_\omega & j_v^2 + j_{\dot\varphi}^2 & -j_\omega j_{\dot\varphi} \\ -j_v j_{\dot\varphi} & -j_\omega j_{\dot\varphi} & j_v^2 + j_\omega^2 \end{bmatrix}$$

B. Safe Secondary Task: Taught Path Following

In the safe context, since path deformation is not necessary, the robot must follow at best the 3D taught path. Thus, as it was during teaching, the camera pan should point forward (i.e, the pan angle should be set to 0, with null angular speed at the equilibrium). To maximize visibility, the linear velocity should be reduced when the path curvature is large. These two requirements can be formulated by using:

$$\mathbf{s}_s = \begin{bmatrix} \int_0^t v(t)dt \\ \varphi \end{bmatrix}, \qquad \hat{\mathbf{s}}_s = \begin{bmatrix} \int_0^t v(t)dt \\ 0 \end{bmatrix}, \qquad \dot{\hat{\mathbf{s}}}_s = \begin{bmatrix} v_s \\ 0 \end{bmatrix}$$

which, replaced in (10), yields:

$$\dot{\mathbf{s}}_s^* = \begin{bmatrix} v_s \\ -\lambda_2 \varphi \end{bmatrix}$$

Since deriving s_s yields $[v\ \dot\varphi]^\top$, the corresponding secondary task Jacobian is:

$$\mathbf{J}_s = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and we obtain:

$$(\mathbf{J}_s\mathbf{P}_1)^+ = \begin{bmatrix} 1 & 0 \\ -\frac{j_v}{j_\omega} & -\frac{j_{\dot\varphi}}{j_\omega} \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{J}_s\mathbf{J}_1{}^+ = \frac{[j_v\ j_{\dot\varphi}]^\top}{j_v^2 + j_\omega^2 + j_{\dot\varphi}^2}$$

C. Secondary Task In The Dangerous Context: Safety

In the dangerous context, the secondary task consists of aligning the robot with f (i.e., shifting its heading by α, with null angular speed at the equilibrium), and reducing the linear velocity, if the obstacles are 'too' near. Denoting with θ the current robot orientation, this is done by using:

$$\mathbf{s}_d = \begin{bmatrix} \int_0^t v(t)dt \\ \theta \end{bmatrix}, \qquad \hat{\mathbf{s}}_d = \begin{bmatrix} \int_0^t v(t)dt \\ \theta + \alpha \end{bmatrix}, \qquad \dot{\hat{\mathbf{s}}}_d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which, replaced in (10), yields:

$$\dot{\mathbf{s}}_d^* = \begin{bmatrix} 0 \\ \lambda_2 \alpha \end{bmatrix}$$

Since deriving s_d yields $[v\ \omega]^\top$, the corresponding secondary task Jacobian is:

$$\mathbf{J}_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and we obtain:

$$(\mathbf{J}_d\mathbf{P}_1)^+ = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{j_v}{j_{\dot\varphi}} & -\frac{j_\omega}{j_{\dot\varphi}} \end{bmatrix}, \qquad \mathbf{J}_d\mathbf{J}_1{}^+ = \frac{[j_v\ j_\omega]^\top}{j_v^2 + j_\omega^2 + j_{\dot\varphi}^2}$$

D. Complete Control Design

Let us now use the variables defined above, to instantiate our controller (9) for visual navigation:

$$\begin{cases} v = (1-H)\, v_s \\ \omega = (1-H)\, \frac{\lambda_1(\hat{x}-x) - j_v v_s + \lambda_2 j_{\dot\varphi}\varphi}{j_\omega} + H\lambda_2\alpha \\ \dot\varphi = H\, \frac{\lambda_1(\hat{x}-x) - \lambda_2 j_\omega \alpha}{j_{\dot\varphi}} - (1-H)\, \lambda_2 \varphi \end{cases} \tag{14}$$

This control law has the following nice properties.

1) In the *safe* context ($H = 0$), (14) becomes:

$$\begin{cases} v = v_s \\ \omega = \frac{\lambda_1(\hat{x}-x) - j_v v_s + \lambda_2 j_{\dot\varphi}\varphi}{j_\omega} \\ \dot\varphi = -\lambda_2 \varphi \end{cases} \tag{15}$$

The image error is controlled only by ω, as in [14] and [28], where obstacles were not considered. The secondary task of maximal visibility (6), instead, is achieved by inputs v and $\dot\varphi$: the linear velocity is reduced to improve tracking according to (11), while the camera is driven forward, to $\varphi = 0$.

2) In the *dangerous* context ($H \approx 1$), a good approximation of (14) is:

$$\begin{cases} v = (1-H)\, v_s \\ \omega = \lambda_2 \alpha \\ \dot\varphi = \frac{\lambda_1(\hat{x}-x) - \lambda_2 j_\omega \alpha}{j_{\dot\varphi}} \end{cases}$$

In this case, the primary task is executed by $\dot\varphi$, while the secondary task of safety (7-8) is controlled by the 2 other degrees of freedom: the linear velocity is reduced (and even zeroed, for $H = 1$), while the angular velocity aligns the robot with f. The camera velocity compensates the robot rotation, to keep the features in view. Note that, for large Z, and small image error, $\dot\varphi \approx -\omega$ (since, from (13), $j_\omega \approx j_{\dot\varphi}$), which is an expected behavior.

3) Since $j_{\dot\varphi} \neq 0$, $\forall x \in I\!R$, for (14) to be well defined when the centroid is visible, it is sufficient that $j_\omega \neq 0$, which is guaranteed, since $|\varphi| \leq \frac{\pi}{2}$, by choosing $Z > \frac{\delta}{2}$, as we have proved in [1]. Moreover, in [28], we have shown that overestimating Z with respect to its real value, is more effective than underestimating it, in terms of navigation performance.

VI. EXPERIMENTAL RESULTS

Experiments have been carried out on a CyCab vehicle, set in car-like mode (i.e., using only the front wheels for steering). The robot is equipped with a coarsely calibrated 70° field of view, B&W camera mounted on a TRACLabs Biclops Pan/Tilt head (the tilt angle is null, to keep the optical axis parallel to the ground), and with a 4-layer, 110° scanning angle, laser SICK LD-MRS. The offset between R and C is $\delta = 0.7$ m, and we set $Z = 15$ m, to prevent singularities in (14). We build the grid by projecting the laser readings from the 4 layers on the ground, and limiting its size to 1.5 m on each side, and 5 m in front. The linear velocity is designed with $V = 0.4$ ms^{-1} and $\gamma = 225$. The velocity has been limited for



Fig. 3. Top to bottom: experiments 1 to 3. Left: reconstruction of the replayed path with (red) and without (yellow) obstacles. Right: snapshots (above) and grids (below) in the presence of obstacles. The red vectors show the orientation of f.

security reasons, and because of the low camera frequency (10 Hz). At this speed, since the scanner acquires at 40 Hz and $K = 20$, neglecting the effect of motion on the occupancies is reasonable. We set: $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, $\kappa = 0.25$, $\rho = 3$ and $\varrho = 4$.

Control scheme (14) has been validated in the three experiments, which are shown in Fig. 3, as well as in the video at: www.irisa.fr/lagadic/demo/demo-cycab-vis-navigation/obstAv.avi.

In each case, after having manually driven CyCab to acquire the image database, two setups (safe and dangerous) are tested. First, no obstacle is present: since H is constantly null, (15) is used, with fixed forward-looking camera ($\varphi = 0$). Then, some obstacles (including pedestrians) are present, near and on the taught path, and the robot must deal with them, during navigation; in addition to endangering the robot, the obstacles may occlude the features. The replayed paths, estimated from odometry and snapshots, are outlined on the left in Fig. 3, both in the safe (yellow) and dangerous (red) setup. Since the focus here is on obstacle avoidance, and accuracy of obstacle-free navigation has been proved in [14], the taught path is not drawn. Giving a general interpretation of the results is hard, since many factors intervene (e.g., obstacles and visual features positions). Nevertheless, in all the experiments, the robot follows the path without colliding, and in spite of occlusions (although a minimal number of visual features is obviously required). Although some portions of the replayed 3D path are far from the taught one, these motions are indispensable to avoid collisions. In the following, we detail the experiments.

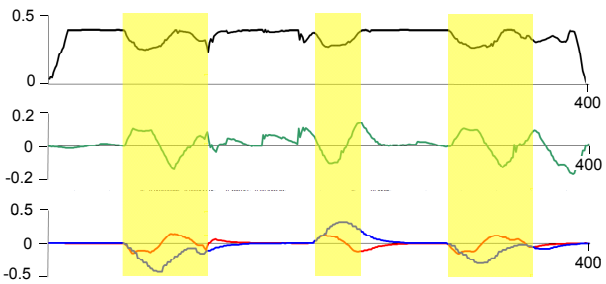


Fig. 4. Relevant variables in experiment 1: v (black, in ms^{-1}), ω (green, in rad s^{-1}), φ (blue, in rad) and $\dot{\varphi}$ (red, in rad s^{-1}). The iterations with non-null H are highlighted in yellow.

The first path (used in experiment 1) is 60 m long, and composed of $N = 34$ key images. Three obstacles are on the path, and the robot is able to overtake them all (the first and third on the left, the second on the right), and reach the last key image. For the dangerous setup, snapshots are shown on top of Fig. 3, and v, ω, $\dot{\varphi}$ and φ are plotted[4] in Fig. 4. The iterations, with $H > 0$ are highlighted in yellow. CyCab is initially near a wall parallel to the Z' axis, which is too far to be considered in the grid. This shows the utility of ignoring lateral data, which would have made the vehicle curve farther from the path. The first obstacle is detected at iteration 60: the vector field f generates a positive rotation (green in Fig. 4), compensated by the camera pan, and v is reduced. The obstacle is overtaken while maintaining feature visibility. Then, since the path is free, the pan angle (blue) is reset to 0 , until the second obstacle detection, which triggers a negative ω, and positive $\dot{\varphi}$. Afterwards, H is activated again by the third obstacle, which is overtaken on the left (iteration 300), while decelerating. The activation function is canceled at iteration 360, and the variables are then driven by (15). The average vehicle velocity v, is slightly reduced, from 0.39 ms^{-1} in the safe setup, to 0.35 in the presence of obstacles. The image error with respect to the visual database, averaged over the

[4]The smooth trend of v at the beginning and end of the 3 experiments is due to the acceleration saturation that is carried out at the control low-level.

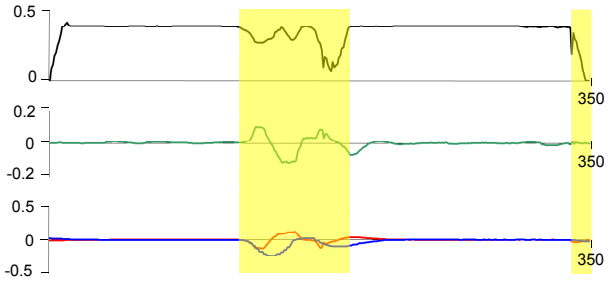


Fig. 5. Relevant variables in experiment 2: v (black, in ms^{-1}), ω (green, in rad s^{-1}), φ (blue, in rad) and $\dot{\varphi}$ (red, in rad s^{-1}). The iterations with non-null H are highlighted in yellow.

experiment, increases from 11 to 17 pixels.

Experiments 2 and 3 are carried out on a path of 50 m and $N = 30$ key images. The path is straight, and there are many cars parked on the street.

In experiment 2, a gray car is parked on the path, and a barrier blocks the road halfway through. Snapshots of the experiment are shown in the center of Fig. 3, and the velocities are plotted in Fig. 5. The car is overtaken on the left at iterations $120-190$, while the robot slows down (black curve). Then, as soon as the street is free again, the robot accelerates and returns on the path, while the camera is reset forward. The barrier is more difficult to deal with. In fact, when detected, it is centered on the Z' axis and orthogonal to it (see grids on the right of Fig. 3); this induces $\alpha = 0$ and drives the robot towards the barrier. However, as the CyCab approaches the barrier, the norm of vector field f increases, and eventually becomes greater than ϱ, to make $H = 1$ and stop the robot. Note that when there is no obstacle, $v \approx V$, because, in contrast with experiment 1, the taught path here is straight, leading to $v_s \approx V$, from (11). The value of v, averaged over the path diminishes, from 0.39 ms^{-1} in the safe setup, to 0.34 in the dangerous one. This time, the average image error does not vary in the two setups (13 pixels in both cases).

In experiment 3, although the gray car and barrier are not present, three persons are standing on the path. On Fig. 3 (bottom), we show some snapshots of the experiment, while the velocities are plotted in Fig. 6. From iteration 50 to 120, the activation function is triggered by the detection of the pedestrians. This generates a counterclockwise angular velocity (the persons are overtaken on the left), and a slight deceleration. Afterwards, since the street is free again, the robot accelerates and returns on the path, which this time is completed up to the last key image, since the barrier has been removed. Just like in experiment 2, when there is no obstacle, $v \approx V$, since the path is straight. The average linear velocity ($v = 0.38$ ms^{-1}), is almost the same as in the safe setup (0.39), and so is the average image error (12 pixels).

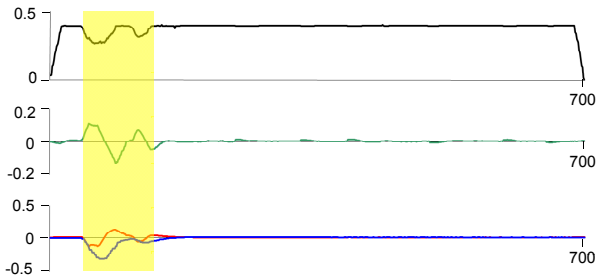


Fig. 6. Relevant variables in experiment 3: v (black, in ms^{-1}), ω (green, in rad s^{-1}), φ (blue, in rad) and $\dot{\varphi}$ (red, in rad s^{-1}). The iterations with non-null H are highlighted in yellow.

VII. CONCLUSIONS

We have designed and implemented a framework with simultaneous obstacle avoidance and vision-based navigation for an outdoor vehicle. It merges techniques from redundancy, potential fields, and visual servoing, and guarantees visual navigation, obstacle bypassing, and collision avoidance by deceleration. The method is validated in outdoor experiments, with real obstacles, such as parked cars and pedestrians. In the future, we plan to take into account moving obstacles, as well as visual occlusions provoked by the obstacles. We also plan to test other obstacle modeling techniques, such as the 'tentacles' presented in [31].

ACKNOWLEDGMENTS

This work was funded by the ANR CityVIP project.

REFERENCES

[1] A. Cherubini and F. Chaumette, "A redundancy-based approach for obstacle avoidance in mobile robot navigation", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010.

[2] O. Gusikhin, N. Rychtyckyj and D. Filev, "Intelligent systems in the automotive industry: applications and trends", in *Knowledge and Information Systems*, vol. 12, no. 2, 2007, pp. 147–168.

[3] M. Buehler, K. Lagnemma and S. Singh (Editors), "Special Issue on the 2007 DARPA Urban Challenge, Part I-III", in *Journal of Field Robotics*, vol. 25, no. 8–10, 2008, pp. 423–860.

[4] Y. Zhang, W. C. Lin and Y.-K. S.Chin, "Driving skill characterization: A feasibility study", *IEEE Int. Conf. on Robotics and Automation*, 2008.

[5] U. Nunes, C. Laugier and M. Trivedi, "Introducing Perception, Planning, and Navigation for Intelligent Vehicles" in *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 3, 2009, pp. 375–379.

[6] A. Broggi, L. Bombini, S. Cattani, P. Cerri and R. I. Fedriga, "Sensing requirements for a 13000 km intercontinental autonomous drive", IEEE Intelligent Vehicles Symposium, 2010.

[7] K. Yamaguchi, T. Kato and Y. Ninomiya, "Vehicle Ego-Motion Estimation and Moving Object Detection using a Monocular Camera", *Int. Conf. on Pattern Recognition*, 2006.

[8] X. Zhou and P. Angelov, "Autonomous Visual Self-localization in Completely Unknown Environment using Evolving Fuzzy Rule-based Classifier", *IEEE Symposium on Computational Intelligence in Security and Defense Applications*, 2007.

[9] D. Schleicher, L. M. Bergasa, M. Ocana, R. Barea and M. E. Lopez, "Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion", in *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 3, 2009, pp. 440–452.

[10] F. Bonin-Font, A. Ortiz and G. Oliver, "Visual navigation for mobile robots: a survey", *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, 2008, pp. 263–296.

[11] J. Minguez, F. Lamiraux and J. P. Laumond, "Motion planning and obstacle avoidance", in *Springer Handbook of Robotics*, B. Siciliano, O. Khatib (Eds.), Springer, 2008, pp. 827–852.

[12] T. Wada, S. Doi and S. Hiraoka, "A deceleration control method of automobile for collision avoidance based on driver's perceptual risk", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009.

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *IEEE Int. Conf. on Robotics and Automation*, 1985.

[14] S. Šegvić, A. Remazeilles, A. Diosi and F. Chaumette, "A mapping and localization framework for scalable appearance-based navigation", *Computer Vision and Image Understanding*, vol. 113, no. 2, 2008, pp. 172–187.
[15] A. A. Argyros, K. E. Bekris, S. C. Orphanoudakis and L. E. Kavraki, "Robot homing by exploiting panoramic vision", *Autonomous Robots*, vol. 19, no. 1, pp. 7-25, 2005.
[16] E. Royer, M. Lhuillier, M. Dhome and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation", *Int. Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.
[17] O. Booij, B. Terwijn, Z. Zivkovic and B. Kröse, "Navigation using an appearance based topological map", *IEEE Int. Conf. on Robotics and Automation*, 2007.
[18] A. Ohya, A. Kosaka and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using a single-camera vision and ultrasonic sensing", *IEEE Trans. on Robotics and Automation*, vol. 14, no. 6, 1998, pp. 969–978.
[19] Z. Yan, X. Xiaodong, P. Xuejun and W. Wei, "Mobile robot indoor navigation using laser range finder and monocular vision", *IEEE Int. Conf. on Robotics, Intelligent Systems and Signal Processing*, 2003.
[20] L. Lapierre, R. Zapata and P. Lepinay, "Simultaneous path following and obstacle avoidance control of a unicycle-type robot", *IEEE Int. Conf. on Robotics and Automation*, 2007.
[21] F. Lamiraux and O. Lefebvre, "Sensor-based trajectory deformation: application to reactive navigation of nonholonomic robots", in *Visual Servoing via Advanced Numerical Methods*, G. Chesi, K. Hashimoto (Eds.), Springer LNCIS Series, 2010, pp. 315–334.
[22] D. Folio and V. Cadenat, "A redundancy-based scheme to perform safe vision-based tasks amidst obstacles", *IEEE Int. Conf. on Robotics and Biomimetics*, 2006.
[23] S. Chiaverini, G. Oriolo and I. D. Walker, "Kinematically redundant manipulators", in *Springer Handbook of Robotics*, B. Siciliano, O. Khatib (Eds.), Springer, 2008, pp. 245–268.
[24] T. Kato, Y. Ninomiya and I. Masaki, "An obstacle detection method by fusion of radar and motion stereo", in *IEEE Trans. on Intelligent Transportation Systems*, vol. 3, no. 3, 2002, pp. 182–188.
[25] C. Harris and M. Stephens, "A combined corner and edge detector", *4th Alvey Vision Conference*, 1988.
[26] J. Shi and C. Tomasi, "Good features to track", *IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.
[27] D. G. Lowe. "Distinctive image features from scale-invariant keypoints", *Int. Journal of Computer Vision*, vol. 60, no. 2, 2004, pp. 91–110.
[28] A. Cherubini, M. Colafrancesco, G. Oriolo, L. Freda and F. Chaumette, "Comparing appearance-based controllers for nonholonomic navigation from a visual memory", *ICRA Workshop on safe navigation in open and dynamic environments*, 2009.
[29] H. Hanafusa, T. Yoshikawa and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy", *IFAC World Congress*, 1981.
[30] A. De Luca and G. Oriolo, "Local Incremental Planning for Nonholonomic Mobile Robots", *IEEE Int. Conf. on Robotics and Automation*, 1994.
[31] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller and H.-J. Wuensche, "Driving with tentacles - Integral structures of sensing and motion", in *Journal of Field Robotics*, vol. 25, no. 9, 2008, pp. 640–673.