ARTICLE IN PRESS

Available online at www.sciencedirect.com
ScienceDirect

Robotics and Autonomous Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Image-based robot navigation from an image memory

Anthony Remazeilles*, François Chaumette

IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

Received 28 November 2005; received in revised form 2 October 2006; accepted 9 October 2006

Abstract

This paper addresses the problem of vision-based navigation and proposes an original control law to perform such navigation. The overall approach is based on an appearance-based representation of the environment, where the scene is directly defined in the sensor space by a database of images acquired during a learning phase. Within this context, a path to follow is described by a set of images, or *image path* extracted from the database. This image path is designed so as to provide enough information to control the robotic system. The central contribution of this paper is the closed-loop control law that drives the robot to its desired position using this image path. This control does not require either a global $3D$ reconstruction or a temporal planning step. Furthermore, the robot is not constrained to converge directly upon each image of the path, but chooses its trajectory automatically. We propose a process of *qualitative visual servoing*, enabling us to enlarge the convergence space towards positioning in a range within a confidence interval. We propose and use specific visual features which ensure that the robot navigates within the visibility path. Experimental simulations are given to show the effectiveness of this method for controlling the motion of a camera in three-dimensional environments (free-flying camera, or camera moving on a plane). In addition, experiments realized with a robotic arm observing a planar scene are also presented.

Keywords: Robotics; Vision-based navigation; Visual servoing

1. Introduction

A robotic system performing a navigation task must have the ability to move itself from an initial position to a desired one. The difficulty of this problem is that these two positions can be far apart. When considering a robotic system with exteroceptive sensors, this particularity means that the information describing the initial position can be totally different and without any relation to the sensor information the robot might obtain at the desired position.

It is thus obvious that the robotic system needs a representation of its environment to perform such a task. In order to realize an autonomous system, this representation should provide enough information for localizing initial and desired positions, defining a path between these two positions, and controlling the motion of the robot during the navigation.

Different types of navigation space description have been proposed in the literature. The most widespread are the ones used for model-based navigation, and appearance-based navigation, which we briefly recall now.

1.1. Model-based approach

Model-based approaches rely on the knowledge of a $3D$ model of the navigation space. The localization is then performed by matching the global model with a local model deduced from sensor data. Features used can be either lines [5], planes [4] or points [3,22]. If the model is not known, a learning step is used for estimating it. The robot is generally controlled by a human operator, like in [22], where the reconstruction is performed using a hierarchical bundle adjustment, or like in [3] where odometry is coupled with a visual tracking system for estimating spherical feature coordinates. A large portion of the article proposes an autonomous mapping of the environment (methods known as SLAM, for *Simultaneous Localization And Mapping*) [26,23,6]. In this case, autonomous motions are performed for discovering new areas, but not for reaching a particular desired position.

* Corresponding address: INSA-IRISA, Lagadic group, Campus Universitaire de Beaulieu, 35042 Rennes, France. Tel.: +33 2 99 84 74 88; fax: +33 2 99 84 71 71.

E-mail addresses: Anthony.Remazeilles@irisa.fr (A. Remazeilles), Francois.Chaumette@irisa.fr (F. Chaumette).

0921-8890/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2006.10.002

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

Once the current robot position is estimated, its motion is generally performed by attracting it towards intermediary desired positions. In [22], motion is deduced from the error measured between the current robot position and the one associated with an intermediary view. In [19,3], the motion is obtained by imposing the features to follow based on the image trajectories observed during the learning step.

1.2. Appearance-based approach

The appearance-based approach (known also as the topological approach) does not require a $3D$ model of the environment. It presents the advantage of working directly in the sensor space. In this case, the environment is described by a topological graph. Each node corresponds to a description of a place in the environment obtained using sensor data, and a link between two nodes defines the possibility for the robot to move autonomously between the two associated positions.

When considering a vision sensor, which is the case in this article, sensor descriptions correspond to images acquired by the camera during the learning step. Localization is usually performed by computing a similarity score between the view acquired by the camera and the different images of the database. This similarity can be based on global descriptors, like the whole image [13,17], color histograms [28], or image gradient [7,14]). Another method consists of taking advantage of image retrieval principles to localize the robot, by using local descriptors, like photometric invariants [21] or SIFT points [15].

Different strategies are then proposed to control the robot during the navigation. In [13,17], a particular motion is associated with each image of the database. At each iteration, the robot performs the motion associated with the closest view of the sequence. However this scheme cannot take into account a potential deviation from the pre-taught path, which can be problematic. In [1,2], the robot converges, using a visual servoing loop, towards each intermediary image of the path, reducing the error measured between the current and the successive desired positions of visual landmarks in the image. However, this approach requires a database precise enough to get satisfying trajectories wherever the initial and desired positions are. Furthermore, it can be considered as useless to converge towards each intermediary position, as long as these local convergences do not culminate in reaching the desired position.

1.3. Approach proposed: A qualitative topological navigation

The work presented here belongs to the second family. We believe that getting rid of a global $3D$ reconstruction and an absolute pose estimation (as needed in model-based approaches) can avoid a potential error propagation while nevertheless merging all the information available in a common $3D$ frame.

Figs. 1 and 2 present in a general way the different processes that enable the system to define a topological path for reaching a particular position. The first figure illustrates

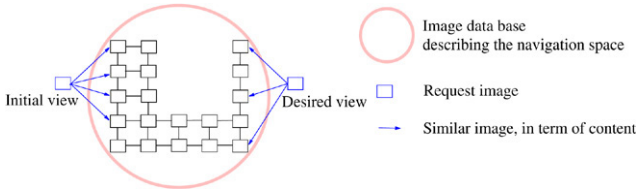

Fig. 1. Qualitative localization of the robotic system by image retrieval. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

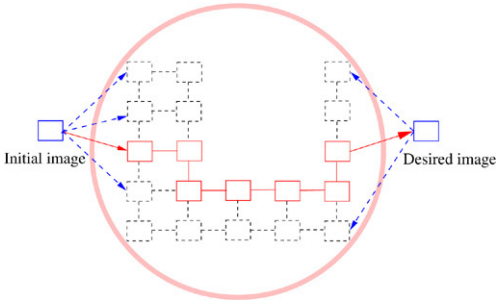

Fig. 2. Image path finding. The robot will use this path to control its motion during the navigation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

how the localization is performed, before the beginning of the motion. During this step, no assumption is made about the robotic system's position. One can note that it is not a particular hypothesis, which can also be found in a large set of works on localization [14,28]. The only information used corresponds to the set of images acquired during an off-line step (this database is surrounded by a circle on the figure). The localization consists in finding those views of the database that are similar, in terms of content, to the request images, either the initial one, or the desired one. On the figure, blue arrows describe the similar views that are found. In [20], we have proposed to use image retrieval schemes to perform this operation. Note that this localization is *qualitative*. Indeed, the $3D$ position of the robotic system is not searched; only the most similar views are.

Once the initial and desired images have been put in relation with some views from the database, the next step consists in reducing the whole database to a set of images describing the area in which the robot is controlled to move. This is illustrated on Fig. 2. This subset of views is directly deduced from the structure of the database. Indeed, like every topological approach, the different views describing the environment are organized within a graph. Each node represents an image, and an edge between two nodes means that the robot can move autonomously between the two associated views. Similar images obtained in the previous steps enable it to incorporate the initial and desired images into the topological graph. Then, the selection of the set of images describing the area in which the robot will navigate is nothing but a search of a path in the graph (this *image* path is illustrated in red on the Fig. 2).

In the following, it is supposed that an image path is provided to the robotic system. The originality of the scheme proposed is that the robotic system is not obliged to converge

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

towards each intermediary position associated to the different images of the path, which gives to it more flexibility during the navigation. The navigation scheme is based on visual servoing. We propose a new qualitative approach in which the visual features used for controlling the system are regulated toward confidence intervals rather than specific desired values.

The next section deals with the navigation scheme. Section 3 presents some experimental results, obtained in simulation and with a real robotic system, which demonstrate the validity of the proposed navigation scheme. Finally, Section 4 contains some concluding remarks.

2. Robot motion control with qualitative visual servoing

In the following, the image path extracted from the database is noted $\psi_0, \ldots, \psi_N$. ψ_0 is the image given by the camera before the beginning of the motion (the initial image). ψ_N is the image that the camera should acquire when the robot reaches its desired position. Features used during the navigation are Harris points [10], matched between the consecutive images of the path. Methods like [27] can be used to determine these correspondences. In the following, $\mathcal{M}_i$ corresponds to the set of points $({}^i\mathbf{x}_j, {}^{i+1}\mathbf{x}_j)$ that are matched between views ψ_i and ψ_{i+1} of the path.

2.1. General control loop

Each set $\mathcal{M}_i$ corresponds to a set of points that are visible between two images of the database. Therefore, these features describe the environment between these two positions. In order to reach the desired position, the robot has to successively go through the places described by the different sets $\mathcal{M}_i$. The navigation task can thus be formulated as follows:

Let $\mathcal{M}_i$ be a set of features matched between views ψ_i and ψ_{i+1} of the image path. Suppose that this set is partially or totally visible within the image frame ψ_t acquired by the camera. Given a set of objective functions describing the image projections of this feature set, the motion of the robotic system aims to make these visual measures reach confidence intervals such that the robot becomes enabled to observe the next set of points $\mathcal{M}_{i+1}$.

It is important to note that moving the robot to observe a set of features does not impose a requirement for it to converge towards every intermediary position in succession. In this formalism, the control law is designed to attract the system into an area in which the visibility is considered as correct.

Fig. 3 shows the general control loop that is used to compute the motion of the robotic system. The different steps involved in this control loop are the following, once a new image has been acquired (this new image being called the *current* one):

1. *Point tracking:* the features ${}^{t-1}\mathbf{x}_j$ visible in the previous view ψ_{t-1} are tracked to obtain their new position in the current view ψ_t.
2. *Point projection update:* features that were not previously considered as visible are transferred from the image path to the current view. It enables the system to determine if new features have gotten inside the camera field of view.

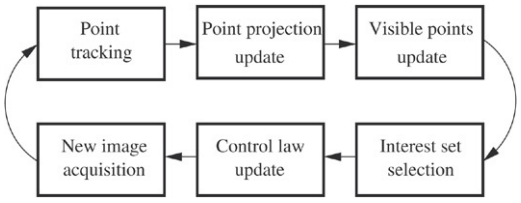

Fig. 3. General control loop used.

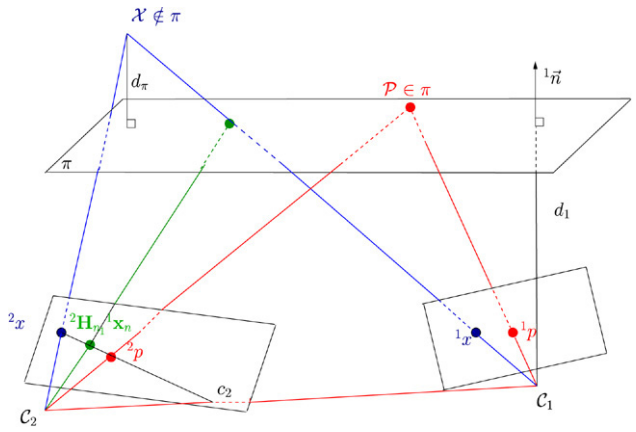

Fig. 4. Relation induced by a homography between two views.

3. *Visible points update:* for all the set of correspondences $\mathcal{M}_i$ defines in the image path, features that are currently projected inside the camera's field of view are recorded, and form the new set of visible points ${}^t\mathbf{x}_j$.
4. *Interest set selection:* amongst all the sets for which some points are already visible, the farthest one is selected. It describes all the features the camera should observe.
5. *Control law update:* Taking into consideration the interest set, the motion of the robot is computed. This motion enables the robot to move towards an area in which the visibility of the whole set is considered as better.

The tracking stage (step 1) can be performed in a real application with a differential point tracker like [12]. The point transfer (step 2) is now described, as well as the step 5 in Section 2.3.

2.2. Geometric relation between images

Let us note ${}^1\mathbf{x}_p$ and ${}^2\mathbf{x}_p$ the projections in two views ψ_1 and ψ_2 of a $3D$ point. These coordinates can be put in relation by the homography ${}^2\mathbf{H}_{p_1}$, through the equation [11]:

$$ {}^2\mathbf{x}_p \propto {}^2\mathbf{H}_{p_1}\,{}^1\mathbf{x}_p + \beta_{1,j}\mathbf{c}_2, \tag{1} $$

with:

$$ {}^2\mathbf{H}_{p_1} = \mathbf{K}\,{}^2\mathbf{H}_{n_1}\,\mathbf{K}^{-1}, \quad {}^2\mathbf{H}_{n_1} = \left({}^2\mathbf{R}_1 + \frac{{}^2\mathbf{t}_1\,{}^1\mathbf{n}^\top}{d_\pi}\right) \quad \text{and} \quad \beta_{1,j} = \frac{d_j}{Z_1 d_\pi}. \tag{2} $$

$\mathbf{K}$ represents the camera's intrinsic parameters, and $\mathbf{c}_2$ the epipole of the second camera. $({}^2\mathbf{R}_1, {}^2\mathbf{t}_1)$ is the rigid motion between the two camera positions. This rotation and

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

translation (up to a scalar factor) can be extracted from the homography [9]. The homography is defined with respect to a reference plane π; $\mathbf{n}$ represents its normal, and d_j the signed distance between the $3D$ point and this plane (see Fig. 4).

If all the points observed belong to the reference plane, only four points are needed for computing the homography [9], and $\beta_{i,j} = 0$. If it is not the case, eight correspondences are needed [16].

The parallax $\beta_{1,j}$ is deduced from the previous equation:

$$\beta_{1,j} = -\frac{\left({}^2\mathbf{H}_{p_1}\,{}^1\mathbf{x}_p \wedge {}^2\mathbf{x}_p\right)^\top \left(\mathbf{c}_2 \wedge {}^2\mathbf{x}_p\right)}{\|\mathbf{c}_2 \wedge {}^2\mathbf{x}_p\|^2}. \tag{3}$$

One can note in Eq. (2) that the parallax term is independent of the second frame position. Nevertheless, as the epipole is only known up to a scalar factor, the Eq. (1) obtained from points data is rather:

$${}^2\mathbf{x}_{p_j} \propto \alpha\ {}^2\mathbf{H}_{p_1}\,{}^1\mathbf{x}_{p_j} + \beta_{\alpha 1_j}\mathbf{c}_2,$$

where $\beta_{\alpha 1_j} = \alpha\beta_{1_j}$. To get rid of this problem, Shashua proposes to scale the homography with respect to a reference point $\mathcal{X}_0 \notin \pi$ [24]:

$${}^2\mathbf{H}'_{p_1} = \frac{\alpha}{\beta_{\alpha 1_0}}\,{}^2\mathbf{H}_{p_1}.$$

By doing this, the parallax becomes invariant to the scalar factor:

$$\beta'_{1_j} = \frac{\beta_{\alpha 1_j}}{\beta_{\alpha 1_0}} = \frac{\alpha d_j}{Z_j d_\pi}\frac{Z_0 d_\pi}{\alpha d_0} = \frac{d_j Z_0}{d_0 Z_j}.$$

Thus, if one knows the homography ${}^3\mathbf{H}_{p_1}$ between the same reference frame ψ_1 and a third image ψ_3, and if this homography is scaled with the same reference point $\mathcal{X}_0$, it is possible to predict the position in ψ_3 of any point matched between views ψ_1 and ψ_2:

$${}^3\mathbf{x}_{p_j} \propto {}^3\mathbf{H}'_{p_1}\,{}^1\mathbf{x}_{p_j} + \beta'_{1,j}\mathbf{c}_3. \tag{4}$$

This principle can be used to perform *image transfer*, between the different images of the path and the current view.

Let us add also that the homography enables us to determine some scene structure information, like the ratio between the depth Z_1 and Z_2 of a $3D$ point [16]:

$$\tau = \frac{Z_2}{Z_1} = \frac{\|[{}^2\mathbf{t}_1]_\times\,{}^2\mathbf{R}_1\,{}^1\mathbf{x}_n\|}{\|[{}^2\mathbf{t}_1]_\times\,{}^2\mathbf{x}_n\|}$$

and the ratio between the depth Z_2 and distance d_1:

$$\rho = \frac{Z_2}{d_1} = \tau\frac{\|{}^2\mathbf{t}_1/d_1\|}{\|{}^2\mathbf{t}_1/Z_1\|}, \tag{5}$$

with ${}^2\mathbf{t}_1/Z_1 = \tau\,{}^2\mathbf{x}_n - {}^2\mathbf{R}_1\,{}^1\mathbf{x}_n$. These relations will be used in the following.

2.3. Computing the control law

2.3.1. Qualitative visual servoing

The new control law we propose can be seen as a *qualitative visual servoing*. Classically, visual servoing is used to minimize an error between a set of visual features $\mathbf{s}$ and their desired values $\mathbf{s}^*$. As $\mathbf{s}$ depends on the camera pose $\mathbf{p}$, the desired pose $\mathbf{p}^*$ is reached when the error measured is null, that is when $\mathbf{s} = \mathbf{s}^*$. For that, a classical control law is given by [8]:

$$\mathbf{v} = -\lambda\mathbf{L}_\mathbf{s}^+(\mathbf{s} - \mathbf{s}^*), \tag{6}$$

where $\mathbf{v}$ is the camera velocity sent to the low-level robot controller, λ is a gain tuning the time-to-convergence of the system, and $\mathbf{L}_\mathbf{s}^+$ is the pseudo inverse of the interaction matrix related to $\mathbf{s}$, which is defined such that $\dot{\mathbf{s}} = \mathbf{L}_\mathbf{s}\mathbf{v}$.

In the method proposed, no particular desired visual features can be defined, since the robot is not required to reach each intermediary pose defined by the image path. The robot is only required to move into areas where the projections of points from set $\mathcal{M}_i$ are considered as satisfactory.

Thus, the robot is only required to reach an area where $\mathbf{s} \in [\mathbf{s}_{\min}; \mathbf{s}_{\max}]$. This is achieved by defining well suited objective functions $\mathcal{V}$, such that their minima correspond to poses where the associated visual feature belongs to $\mathbf{s} \in [\mathbf{s}_{\min}; \mathbf{s}_{\max}]$. Then, the gradients of these functions $\nabla\mathcal{V}(\mathbf{p})$ are used as visual features, replacing $\mathbf{s}$ in Eq. (6). The desired feature $\mathbf{s}^*$ in this equation, is then equivalent to $\nabla\mathcal{V}(\mathbf{p})^*$, which is equal to zero. The control law that is used instead of Eq. (6) is thus:

$$\mathbf{v} = -\lambda\mathbf{L}_{\nabla\mathcal{V}}^+\nabla\mathcal{V}, \tag{7}$$

where $\mathbf{L}_{\nabla\mathcal{V}}$ is the interaction matrix associated to the gradient of $\mathcal{V}$.

Sections 2.3.2–2.3.4 present the different functions $\mathcal{V}(\mathbf{p})$, as well as $\nabla\mathcal{V}(\mathbf{p})$ and $\mathbf{L}_{\nabla\mathcal{V}}$ used in our scheme. Section 2.3.5 finally describes how they are merged together.

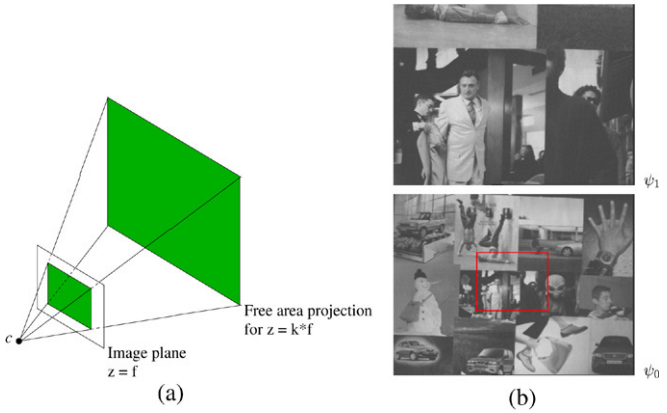

Fig. 5. Motions along the optical axis: (a) visibility cone associated with free area $\mathcal{I}_{\text{free}}$, (b) comparing the image borders after a motion along the optical axis.

2.3.2. Progressing along the path

This first objective function deals with the camera's motions along the optical axis. In the case of a pinhole camera, as illustrated on Fig. 5(a), the projection of a $3D$ point $\mathcal{X} = (X, Y, Z)$ is inversely proportional to Z, since $\mathbf{x} = (X/Z, Y/Z)$. Therefore, the higher Z is, the closer to the image center the point projection is. The same reasoning holds when moving the camera along the optical axis. Indeed, if one considers a motion between two views that is reduced to a

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

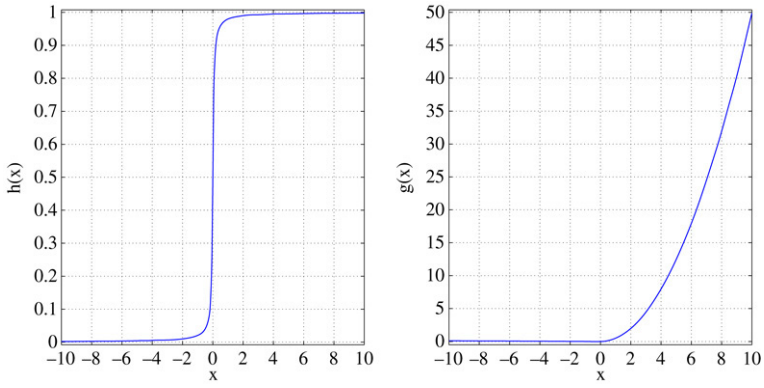

Fig. 6. Functions h and g used to smoothen the objective function (see Eq. (11)).

translation t_z, then the projection of a point becomes, in the second view, $(X/(t_z + Z), Y/(t_z + Z))$. The farther the current camera is to the next image, the closer to the image center is the point projection. This point is illustrated on Fig. 5(b), where ψ_0 is the current view given by the camera, and ψ_1 is the next image from the path. As one can easily see, the area defined by the set of image points is smaller than the one observed in the image ψ_1. This information is here used to consider motions along the optical axis.

To describe the feature projection area, a measure based on centered moments is used. More precisely, it is composed of the second order centered moments:

$$a = \mu_{02} + \mu_{20}.$$

We recall that, for a set of n features, the centered moment μ_{ij} of order $i + j$ is:

$$\mu_{ij} = \sum_{k=0}^{n} (x_k - x_g)^i (y_k - y_g)^j,$$

where (x_g, y_g) is the image center of gravity of the n points ($n = \text{card}(\mathcal{M}_i)$). The closer the points are to the camera, the bigger the value of a is. Intuitively, a is closely related to the area of the set of points in the image. The following measure a_n compares the current value of a with a^*, the one obtained on the next image of the path [25]:

$$a_n = \sqrt{\frac{a^*}{a}}. \tag{8}$$

Since the robot is not required to precisely reach each position associated to the image path, it is not required to obtain the measure a^*, but rather a value sufficiently close to the one measured in the path frame ψ_{i+1}. Let $p \in [0\ \ 1]$ be the percentage of liberty authorized around a_n^*. A satisfactory measure is one such that:

$$a_m = a_n^*(1 - p) < a_n < a_n^*(1 + p) = a_M.$$

This could be controlled with the following function:

$$\mathcal{V}_{a_n}(a_n) = \begin{cases} \frac{1}{2}(a_n - a_M)^2 & \text{if } a_n > a_M \\ \frac{1}{2}(a_m - a_n)^2 & \text{if } a_n < a_m \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

In order to obtain a smooth and continuous transition between the three cases, we propose to use, rather:

$$\mathcal{V}_{a_n}(a_n) = g(a_n - a_M) + g(a_m - a_n), \tag{10}$$

where (see Fig. 6):

$$g(x) = \frac{1}{2}x^2 h_k(x) \quad \text{and} \quad h_k(x) = \frac{\arctan(k\pi x)}{\pi} + \frac{1}{2} \tag{11}$$

$h_k(x)$ is the arc-tangent function normalized on $[0; 1]$. It corresponds to a "*heavy-side*" function which defines a transition between the values 0 and 1. This transition occurs when $x = 0$. The constant scalar k enables us to regulate the curvature of the transition from one value to the other. As it can be seen on Fig. 7, $\mathcal{V}_{a_n}$ is null when the measure a_n belongs to the confidence interval. It tends toward the parabolic function when a_n moves away from this free area.

The error associated with $\mathcal{V}_{a_n}$ is derived as:

$$e_{\nabla_{a_n}} = \nabla_{a_n}\mathcal{V}_{a_n} = \frac{\partial \mathcal{V}_{a_n}}{\partial a_n}, \tag{12}$$

where $\nabla_{a_n}\mathcal{V}_{a_n}$ is:

$$\begin{aligned}\nabla_{a_n}\mathcal{V}_{a_n} = {} & (a_n - a_M)h(a_n - a_M) + \mathcal{O}(a_n - a_M) \\ & + (a_n - a_m)h(a_m - a_n) - \mathcal{O}(a_m - a_n),\end{aligned}$$

in which:

$$\mathcal{O}(x) = \frac{kx^2}{2(1 + k^2\pi^2x^2)}. \tag{13}$$

If one gets rid of the functions h and $\mathcal{O}$ used for continuity matters, this gradient can be approximated by:

$$\nabla_{a_n}\mathcal{V}_{a_n} = \begin{cases} a_n - a_M & \text{if } a_n > a_M \\ a_n - a_m & \text{if } a_n < a_m \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Finally, to derive the control law, the interaction matrix associated with this feature has to be computed. By using Eq. (12), the derivative of $e_{\nabla_{a_n}}$ with respect to time is given by:

$$\dot{e}_{\nabla_{a_n}} = \frac{\partial e_{\nabla_{a_n}}}{\partial a_n}\frac{\mathrm{d}a_n}{\mathrm{d}t} = \frac{\partial^2 \mathcal{V}_{a_n}}{\partial a_n^2}\mathbf{L}_{a_n}\mathbf{v} = \mathbf{L}_{\nabla_{a_n}}\mathbf{v}, \tag{15}$$

where $\mathbf{L}_{a_n}$ is the interaction matrix related to a_n and $\mathbf{L}_{\nabla_{a_n}}$ is the one associated to the visual feature $\nabla_{a_n}\mathcal{V}_{a_n}$. By using the

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

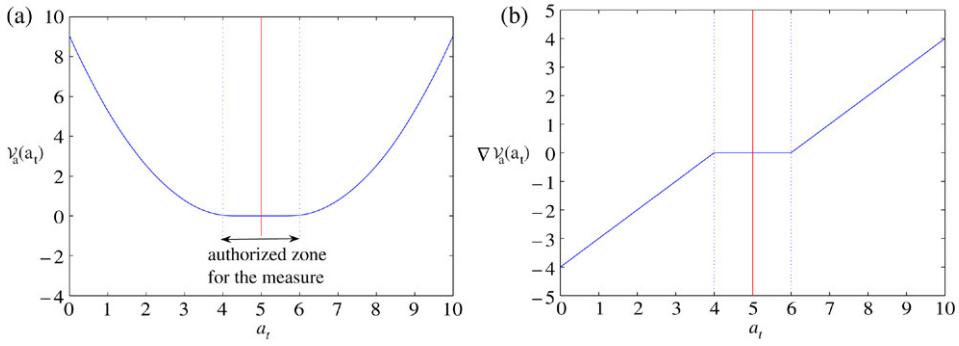

Fig. 7. Controlling the motion along the optical axis: (a) the function used, (b) its gradient.

approximation proposed on Eq. (14), we get:

$$\frac{\partial^2 \mathcal{V}_{a_n}}{\partial a_n^2} = \begin{cases} 1 & \text{if } a_n < a_m, \text{ or } a_n > a_M \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Since $e_{\nabla_{a_n}} = 0$ for $a_m < a_n < a_M$, $\mathbf{L}_{a_n}$ can be chosen as a good approximation of $\mathbf{L}_{\nabla_{a_n}}$. An approximation of this interaction matrix $\mathbf{L}_{a_n}$ is given by Tahri and Chaumette [25]:

$$\mathbf{L}_{a_n} = \begin{bmatrix} 0 & 0 & -\dfrac{1}{Z^*} & -a_n\epsilon_1 & a_n\epsilon_2 & 0 \end{bmatrix}, \tag{17}$$

with:

$$\epsilon_1 = y_g + (y_g\mu_{02} + x_g\mu_{11} + \mu_{21} + \mu_{03})/a$$
$$\epsilon_2 = x_g + (x_g\mu_{20} + y_g\mu_{11} + \mu_{12} + \mu_{30})/a,$$

where ϵ_1 and ϵ_2 can be neglected with respect to 1. This approximation is correct only if the camera is parallel to a planar object (at a distance Z^*). However, as our experiments will show, this approximation does not disturb the results (and we have set $Z^* = 1$).

2.3.3. Feature position control

The next function controls the point projections onto the image plane. It should be clear that all the features of the interest set should project inside the camera field of view.

Feature projection coordinates $\mathbf{x}_j = (x_j, y_j)$ are satisfactory if they are such that: $x_j \in [x_m + \alpha; x_M - \alpha]$ and $y_j \in [y_m + \alpha; y_M - \alpha]$, where x_m, x_M, y_m and y_M are the image borders, and α is a positive constant defining a free projection area $\mathcal{I}_{\text{free}}$ within the image frame (see Fig. 8).

The function $\mathcal{V}_{\mathbf{s}}$ characterizing point projections on the image plane is defined by:

$$\mathcal{V}_{\mathbf{s}} = \sum_j \mathcal{V}_{s_{(\mathbf{x}_j)}} \quad \text{with}$$

$$\mathcal{V}_{s_{(\mathbf{x}_j)}} = g(x_m - x_j) + g(x_j - x_M) + g(y_m - y_j) + g(y_j - y_M),$$

where $g(x)$ has already been given in Eq. (11). Fig. 9 represents this objective function for a single point, and is one component of its gradient. In the application considered here, $\nabla_{\mathbf{s}}^\top \mathcal{V}_{\mathbf{s}}$ gathers

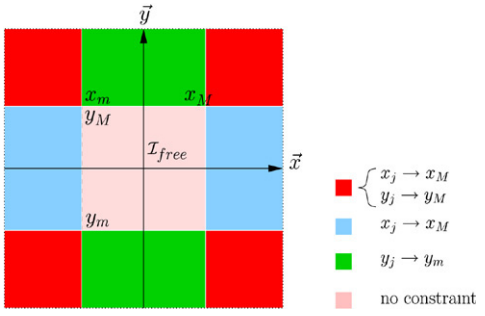

Fig. 8. Areas of activation induced in the control law, displayed on the image plane ($\mathcal{I}_{\text{free}}$ is a restriction of the image frame). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the gradients of the different features of the interest set $\mathcal{M}_i$:

$$\nabla_{\mathbf{s}}^\top \mathcal{V}_{\mathbf{s}} = \left(\nabla_{\mathbf{s}}^\top \mathcal{V}_{\mathbf{s}}(\mathbf{x}_1), \ldots, \nabla_{\mathbf{s}}^\top \mathcal{V}_{\mathbf{s}}(\mathbf{x}_n)\right),$$

where $\nabla_{\mathbf{s}}^\top \mathcal{V}_{\mathbf{s}}(\mathbf{x}_j)$ is given in Box I.

By using the same approximation as before, the interaction matrix related to $\nabla_{\mathbf{s}}^\top \mathcal{V}_{\mathbf{s}}$ is approximated by the interaction matrix $\mathbf{L}_{\mathbf{s}}$ associated with the image point coordinates. This matrix is given by:

$$\mathbf{L}_{\mathbf{s}} = \mathbf{L}(\mathbf{x}, d_{i+1}) = \begin{bmatrix} \dfrac{1}{d_{i+1}}\mathbf{S} & \mathbf{Q} \end{bmatrix},$$

where d_{i+1} is the distance between the image frame ψ_{i+1} and the reference plane π. $\mathbf{S} = (\mathbf{S}_1, \ldots, \mathbf{S}_n)$ and $\mathbf{Q} = (\mathbf{Q}_1, \ldots, \mathbf{Q}_n)$ are two $2n \times 3$ matrices independent to d_{i+1}:

$$\mathbf{S}_j = \frac{1}{\rho_j}\begin{bmatrix} -1 & 0 & x_j \\ 0 & -1 & y_j \end{bmatrix}$$

$$\mathbf{Q}_j = \begin{bmatrix} x_j y_j & -(1 + x_j^2) & y_j \\ 1 + y_j^2 & -x_j y_j & -x_j \end{bmatrix}.$$

Scalar ρ_j is given by Eq. (5), using the homography ${}^t\mathbf{H}_{i+1}$ between the current view ψ_t and the image ψ_{i+1} of the path.

2.3.4. Landmark based on image orientation

The last visual measure deals with the error of orientation that can be measured between the current camera pose and the images of the path. This rotation can be extracted from the

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

$$\nabla_{\mathbf{s}}^{\top}\mathcal{V}_{\mathbf{s}}(\mathbf{x}_j) = \begin{bmatrix} (x_j - x_M)h(x_j - x_M) + (x_j - x_m)h(x_m - x_j) + \mathcal{O}(x_j - x_M) - \mathcal{O}(x_m - x_j) \\ (y_j - y_M)h(y_j - y_M) + (y_j - y_m)h(y_m - y_j) + \mathcal{O}(y_j - y_M) - \mathcal{O}(y_m - y_j) \end{bmatrix},$$

Box I. Gradient of the function $\mathcal{V}_{\mathbf{s}}$ for a given image point position $\mathbf{x}_j$.

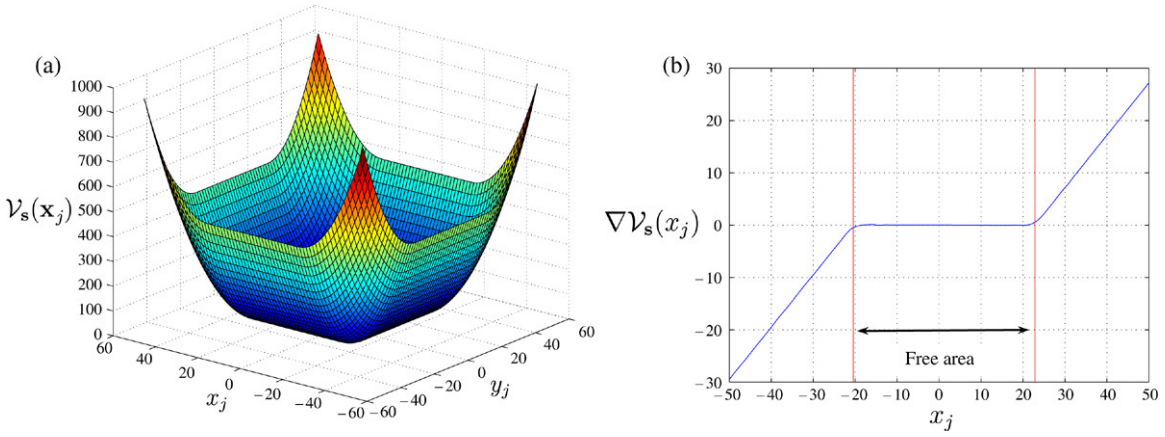

Fig. 9. Function considering feature visibility: (a) function for one point, (b) gradient, for x_j coordinates.

homography ${}^{t}\mathbf{H}_{i+1}$ linking the current view ψ_t with the image ψ_{i+1}.

The minimal representation of the rotation $\theta\mathbf{u}$ is obtained from the coefficients $r_{ij(i=1...3,\,j=1...3)}$ of the matrix ${}^{t}\mathbf{R}_{i+1}$, by using:

$$\theta\mathbf{u} = \frac{1}{2\mathrm{sinc}\,\theta}\begin{pmatrix} r_{32} - r_{23} \\ r_{31} - r_{13} \\ r_{21} - r_{12} \end{pmatrix},$$

with $\theta = \arccos((r_{11}+r_{22}+r_{33}-1)/2)$, and where the cardinal sine $\mathrm{sinc}\,\theta$ is such that $\sin\theta = \theta\,\mathrm{sinc}\,\theta$.

Once again, an interval is used to define the quality of the current orientation:

$$-p_\theta < \theta u_i < p_\theta,$$

where p is a positive scalar belonging to [0 1]. The associated function is:

$$\mathcal{V}_{\theta\mathbf{u}}(\theta u_i) = g(\theta u_i - p_\theta) + g(-p_\theta - \theta u_i), \tag{18}$$

whose corresponding gradient is:

$$\nabla_{\theta\mathbf{u}}\mathcal{V}_{\theta\mathbf{u}}(\theta u_i) = (\theta u_i - p_\theta)h(\theta u_i - p_\theta) + \mathcal{O}(\theta u_i - p_\theta) + (\theta u_i + p_\theta)h(-p_\theta - \theta u_i) - \mathcal{O}(\theta u_i + p_\theta). \tag{19}$$

The interaction matrix of $\nabla_{\theta\mathbf{u}}^{\top}\mathcal{V}_{\theta\mathbf{u}}$ is approximated by $\mathbf{L}_{\theta\mathbf{u}}$ [16]:

$$\mathbf{L}_{\theta\mathbf{u}} = [\mathbf{0}_3\ \mathbf{L}_w], \quad \text{where}$$

$$\mathbf{L}_w = \mathbb{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\mathrm{sinc}\,\theta}{\mathrm{sinc}^2\frac{\theta}{2}}\right)[\mathbf{u}]_\times^2. \tag{20}$$

The function defined here is very similar to the ones defined before; the corresponding curves are therefore not shown. Let us note that in our experiments, this function is only used for controlling rotation around $\vec{x}$ and $\vec{y}$ axis. Indeed, rotations around the optical axis do not improve the feature visibility; nor do they push the robot towards the desired position. But it could be also possible to control this degree of freedom in other applications.

The next subsection presents how these different visual measures are combined to compute the motion of the robotic system.

2.3.5. Control law

Previous subsections have described three different functions, ascribing to each a different constraint on the visual feature configurations, and all these constraints have to be observed simultaneously. This is achieved by stacking the visual features in the control law, which gives:

$$\mathbf{v} = -\lambda\mathbf{L}^{-1}\nabla,$$

where $\mathbf{L}$ and ∇ are respectively a stack of interaction matrices and gradients previously defined:

$$\mathbf{L} = (\mathbf{L}_{\mathbf{s}}, \mathbf{L}_{a_n}, \mathbf{L}_{\theta\mathbf{u}}), \quad \text{and} \quad \nabla = (\nabla_{\mathbf{s}}^{\top}\mathcal{V}_{\mathbf{s}}, \nabla_{a_n}\mathcal{V}_{a_n}, \nabla_{\theta\mathbf{u}}^{\top}\mathcal{V}_{\theta\mathbf{u}}).$$

The gradient $\nabla_{\mathbf{s}}\mathcal{V}_{\mathbf{s}}$ is computed as described in Section 2.3.3. Features ${}^{t}\mathbf{x}_{n_j}$ used correspond to the set $\mathcal{M}_i$, which gathers landmarks matched between the views ψ_i and ψ_{i+1} of the path. Their coordinates are obtained either by the tracking step or by the prediction step.

In $\nabla_{a_n}\mathcal{V}_{a_n}$ (defined in Section 2.3.2), the desired value a^* is based on the feature projections observed in the image ψ_{i+1} from the path. Once again, only features from $\mathcal{M}_i$ are considered. Measure a is deduced from the position of these landmarks in the current image $\mathcal{I}_t$.

Finally, with the homography between current view and view ψ_{i+1}, the rotation ${}^{t}\mathbf{R}_{i+1}$ is extracted, from which is obtained the vectorial representation $\theta\mathbf{u}$, which in turn is used to compute $\nabla_{\theta\mathbf{u}}\mathcal{V}_{\theta\mathbf{u}}$ and $\mathbf{L}_{\theta\mathbf{u}}$ (see Section 2.3.4).

When enough points from the last set $\mathcal{M}_{N-1}$ are visible, the robotic system is in the vicinity of its desired position. A

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

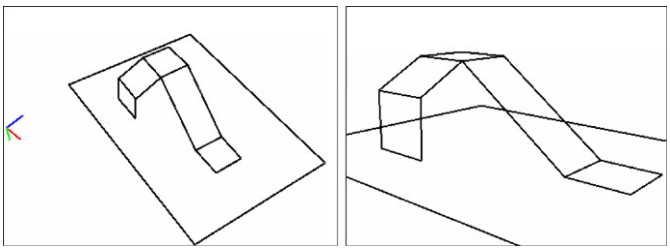

Fig. 10. Views of the 3D object used for simulations. The camera frame is represented as follows: $\vec{x}$ axis in red, $\vec{y}$ axis in green, and $\vec{z}$ axis in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

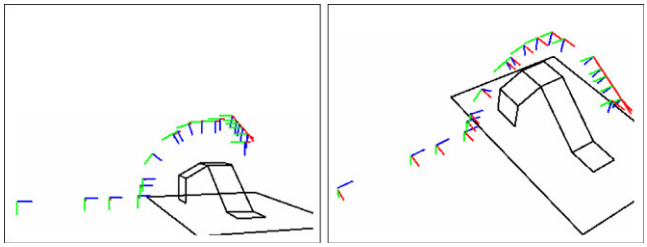

Fig. 11. Exp. 1: Positions associated with the image path.

classical visual servoing scheme can then be used to converge towards this position.

To conclude, the control law proposed can be considered as defining a *qualitative visual servoing* process. The word *qualitative* means that there is not a single position that enables the system to have this convergence. Indeed, contrary to classical visual servoing, no particular desired value is required, but rather a range within a confidence interval.

Furthermore, one can note that the control law proposed merges features expressed directly in the image with information expressed in the configuration space. Merging $2D$ and $3D$ information has also ensured the success of the $2\,1/2D$ visual servoing proposed in [16].

3. Experimental results

In this section, several experimental results are proposed to demonstrate the validity of the control law proposed. In order to study the behavior of this control law without adding potential noise that could bring the tracking and prediction steps, a simulator has been developed. The first subsection presents results for a camera with five degrees of freedom, and the next subsection for a camera moving on a plane as if it was mounted on a holonomic mobile robot. Finally, Section 3.2 presents some experiments realized on a real robotic system, with a planar environment.

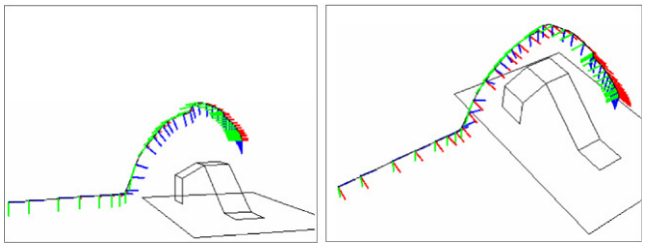

Fig. 13. Exp. 1: Realized trajectory.

3.1. Setup one: Five degrees of freedom camera

First of all, let us consider the object around which the camera will move. It is composed by a set of planes (as shown on Fig. 10). With each face is associated a set of points. All the points defined do not belong to a particular face.

On Fig. 11, a navigation task is presented by the set of positions associated with the image path that is contained in the image database. Some of the corresponding views are presented on Fig. 12. The position of the camera during the navigation is presented on Figs. 13 and 14. On the second figure, the pose of the robotic system (curves) is compared to the positions of the several images from the path (crosses). Vertical lines indicate a change of interest set $\mathcal{M}_i$. If the robot was converging towards every image from the path, the curves would pass through the crosses. This clearly not the case.

As scheduled by the image path, the beginning of the motion is mainly a translation along the optical axis (until iteration 370). When the robot is close enough to the object, translations along $\vec{y}$ axis and rotations around $\vec{x}$ axis enable it to reach the upper part of the object (iterations 370 to 800). Then, translations along $\vec{x}$ axis are performed to reach the desired final area.

In the next example, only the initial position is changed. As shown on Fig. 15, the initial projection of the object is close to the left border of the image. As it can be seen on Fig. 16, the beginning of the motion is mainly a translation along the optical axis (up to iteration 370). Indeed, the second visual measure presented in Section 2.3.3 is used to make the object projection area grow, by getting closer to the object. But at the same time, the object gets closer to the image's borders. The first visual measure is thus used to ensure that the object stays in the camera's field of view, which is achieved thanks to its translation along $\vec{x}$ axis and rotation around $\vec{y}$ axis.

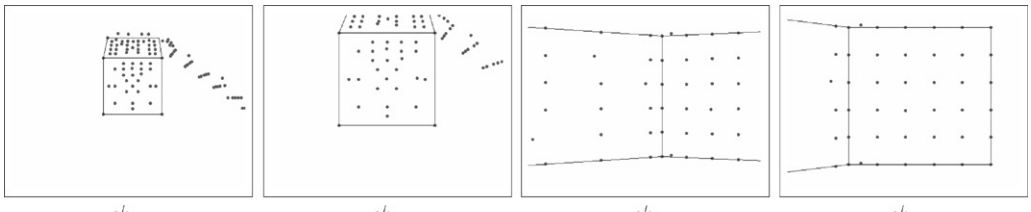

Fig. 12. Exp. 1: Examples of images from the path (ψ_0: initial image, ψ_{19}: desired one).

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

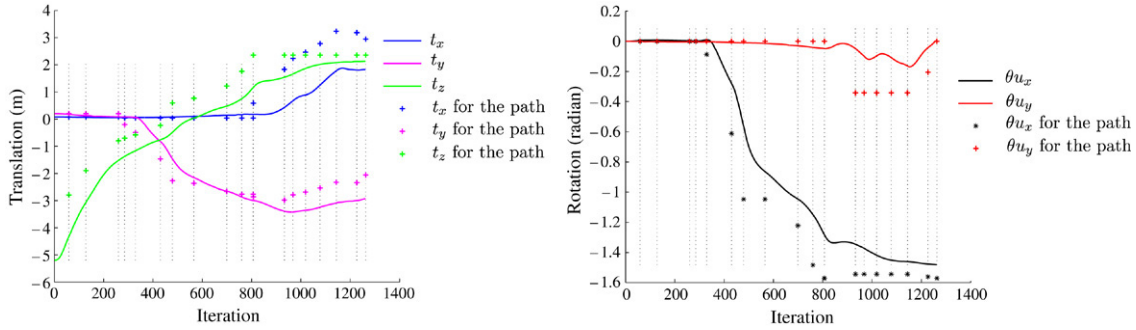

Fig. 14. Exp. 1: Position of the camera (translation and orientation) during the navigation. Vertical lines indicate a change of the interest set $\mathcal{M}_i$ used to control the system. Crosses indicate the pose associated with the view ψ_{i+1} of the set $\mathcal{M}_i$ just before the change of interest set. The robotic system does not converge towards each of these positions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

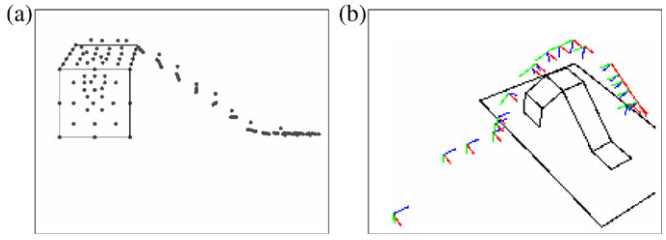

Fig. 15. Exp. 2: (a) initial image, (b) image path positions.

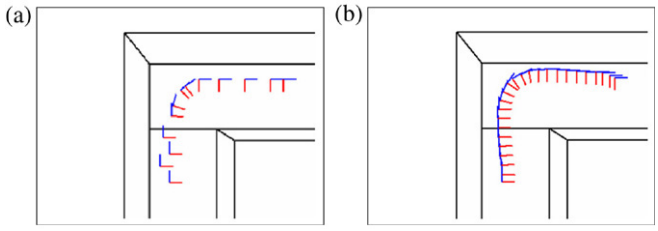

Fig. 17. Exp. 3: Image path positions and the realized trajectory.

3.2. Set up two: Robotic system moving on a plane

In the next experiment, the proposed control law has been used for controlling the motion of a robot moving on a plane. The navigation space corresponds to a corridor, defined by a set of planes, on the floor and on the walls.

The robot is controlled here with two inputs: one for the translation along the $\vec{z}$ axis, and one for the rotation around the $\vec{y}$ axis. The interaction matrices $\mathbf{L}_s$, $\mathbf{L}_{\theta\mathbf{u}}$ and $\mathbf{L}_{a_n}$ are thus simplified to consider only this kind of motion.

Fig. 17(a) represents the image path. One can note in this figure that in the beginning of the image path, some images are not situated on the shortest path. Views ψ_2 and ψ_4 are shifted towards the left. It is obvious here that the robotic system does not need to reach these positions to perform its navigation task.

Figs. 17(b) and 18 describe the trajectory realized by the robotic system. It can be seen that it does not reach the position associated with views ψ_2 and ψ_4. On the second figure, the blue line (robot position on $\vec{x}$ axis), stays far from the crosses 1 and 3.

3.3. Experiments on a robot arm

The next few experiments have been realized on a robot arm with six degrees of freedom with an on-board camera. The navigation space considered is a plane on which several images are stuck. In order to demonstrate the validity of our approach, we select a case where the robot cannot go in a straight path from the initial position to the desired one. Images extracted from the database and defining the path the robot needs to perform are shown in Fig. 19.

When considering a planar scene, the image transfer presented in Section 2.2 is much simpler. Indeed, the relation between the points in two views is reduced to ${}^2\mathbf{x}_p \propto {}^2\mathbf{H}_{p_1}\,{}^1\mathbf{x}_p$, since all the points belong to the reference plane used to compute the homography. Therefore, the information in this matrix is sufficient to perform the point transfer. The

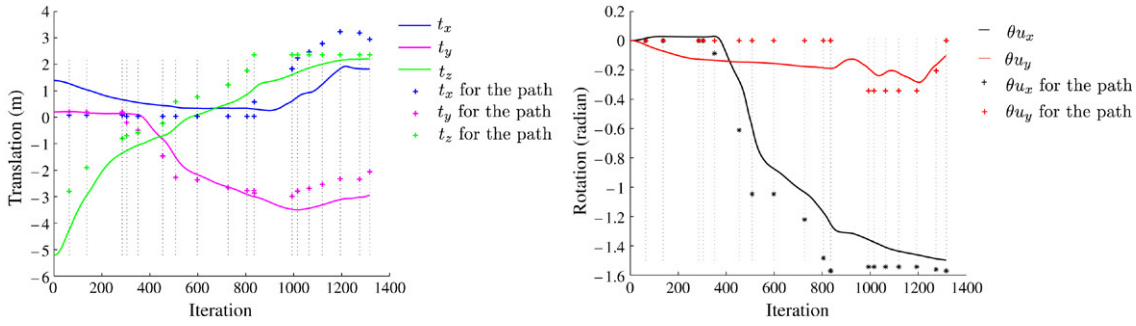

Fig. 16. Exp. 2: Robotic system positions and orientations during navigation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

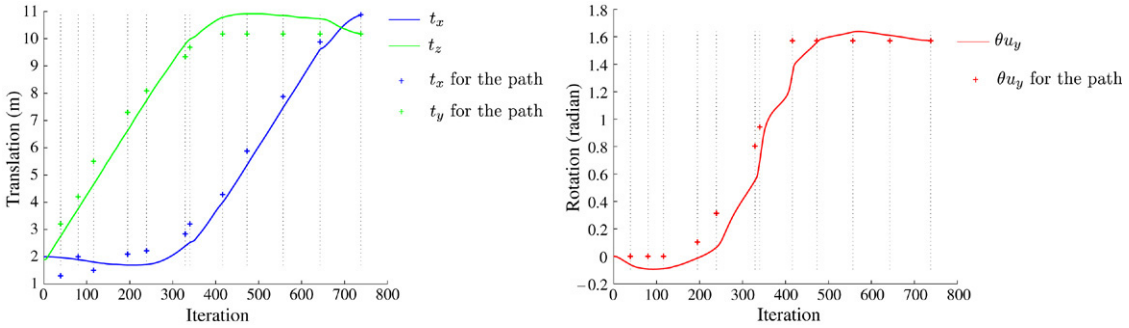

Fig. 18. Exp. 3: Robotic system positions and orientations during navigation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

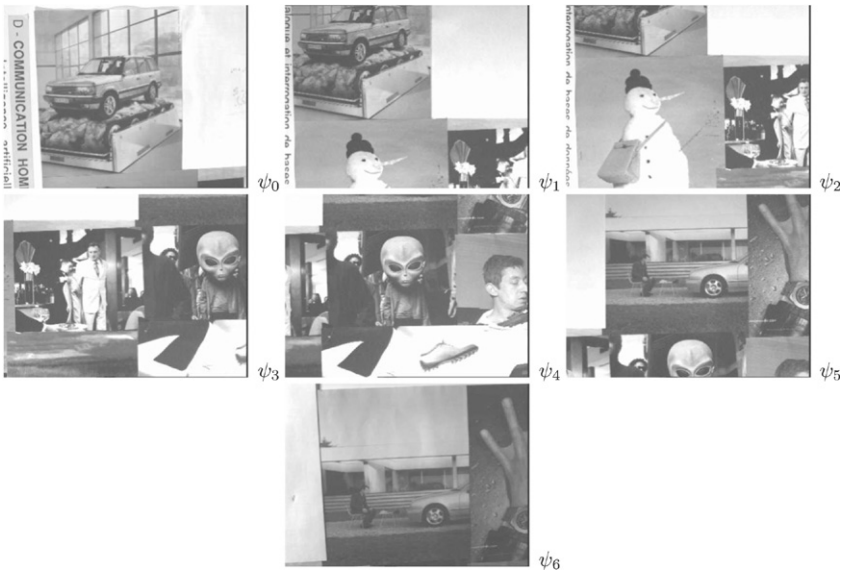

Fig. 19. Exp. 4: Image path used. ψ_0 is the initial image, and ψ_6 is the desired one. Other ones have been automatically extracted from the database.

combination of homographies is also straightforward. Thus, if we consider that the point correspondences between the images ψ_t and ψ_i enable us to compute the associated homography (four matches are sufficient), and if the homography between images of the path ψ_i and ψ_j has been computed, then one can estimate the homography between views ψ_t and ψ_j, which is nothing but:

$$ {}^t\mathbf{H}_{p_j} = {}^t\mathbf{H}_{p_i}\, {}^i\mathbf{H}_{p_j}\,. \tag{21}$$

This principle is used during the navigation for predicting the position of the points that are entering inside the camera's field of view. It has also been used for creating the Fig. 20, in which all the points and image borders of the image path are projected onto the first reference frame. As one can see, all the points are not visible in the first view. Furthermore, their position in the image does not enable the robot to move directly from the initial to the desired position.

Fig. 21 presents the trajectory of the principal point of the camera during the navigation. Ensuring that the points of the next scheduled set $\mathcal{M}_i$ enter the camera's field of view is sufficient to perform the navigation without reaching the pose

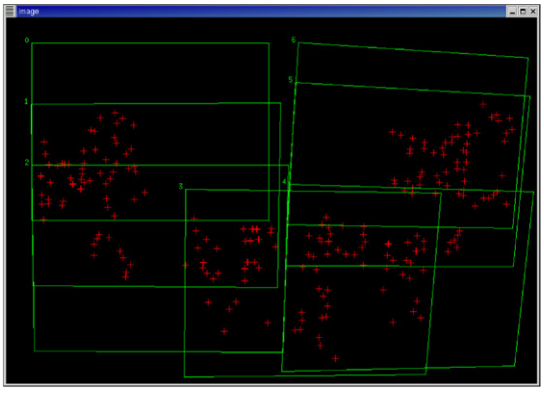

Fig. 20. Exp. 4: Points and image borders projected onto the first image plane.

of each image of the path. Fig. 22 compares the obtained $2D$ trajectory with two other approaches. The first method is an extension of classical image-based visual servoing: the robotic system successively converges towards each image of the path. Once the error measured between the current and desired point positions is sufficiently small, the system considers the next image in the path as its new desired position. In the second

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

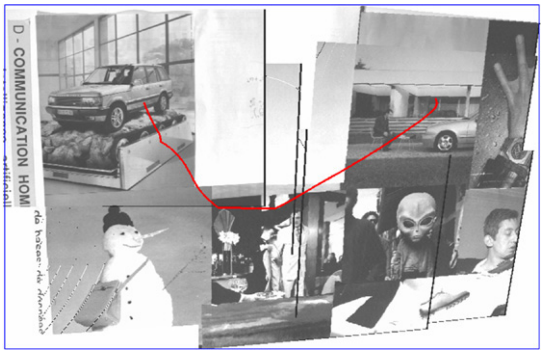

Fig. 21. Exp. 4: Principal point trajectory projected onto the first image plane.

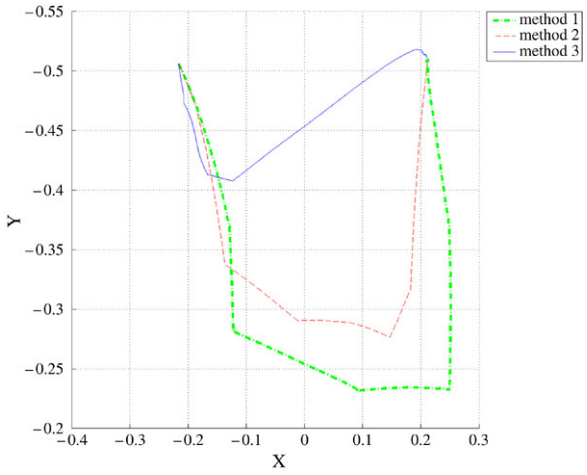

Fig. 22. Exp. 4: Comparison of 2D robot trajectories for the path defined by Fig. 19. Method 1: iterative classical visual servoing with convergence towards each intermediary image. Method 2: the current visual servoing is skipped when enough points from the next view are visible. Method 3: method proposed in this paper. Our approach gives the shortest path. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

method [18], the robot still converges towards intermediary images with an image-based visual servoing, but the current servoing is stopped as soon as enough points from the next image in the path are visible (this information is obtained by performing the point transfer with Eq. (21)). The next image in the path is then considered as the desired one. Therefore, the robot no longer converges towards every image of the sequence (as we can see in Fig. 22), but it is still dependent on the intermediary poses.

As one can see, while ensuring that the robotic system stays in areas where enough points are visible, the method proposed here manages to realize a shorter trajectory. Furthermore, this trajectory is less dependent on the poses of the images from the path. Indeed, in the second objective function $\mathcal{V}_s$ (see 2.3.3), all the points considered are projected on the current image plane, and the measures realized only consider these positions. For the two other approaches, the points are explicitly required to reach the positions measured in the next image of the path. The motion performed is thus naturally dependent on the pose between this view and the current one.

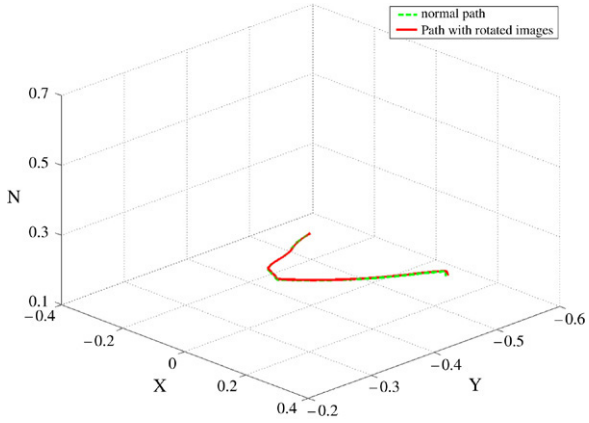

Fig. 23. Exp. 4: Robot trajectories compared (path defined by Fig. 19 and the same with rotated images). The two trajectories are equivalent, since the robotic system uses the feature positions in the current frame to control its motion, and not the ones observed within the images from the path. Rotated images do not disturb the control law. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In the next experiment, a 180° rotation is applied to images ψ_1 and ψ_5 of the path. Navigating this path with the first approach constrains the robot to make these useless rotations during motions $\psi_0 - \psi_1$, $\psi_1 - \psi_2$, $\psi_4 - \psi_5$ and $\psi_5 - \psi_6$. The second method, even if it avoids the convergence towards each of the intermediary images, realizes a part of these rotations anyway. Fig. 23 compares the trajectory obtained with our approach for the path without rotation, and for the trajectory obtained when views ψ_1 and ψ_5 are rotated. The two trajectories are nearly the same. As expected, the rotations around the optical axis do not affect our approach at all.

4. Conclusion

This paper has presented a new control law for robot navigation. An image path is first extracted from a visual memory describing the environment. This image path defines the visual features that the camera should observe during the motion. The control law proposed does not require a $3D$ reconstruction of the environment. Furthermore, the images constituting the path are not considered as successive desired positions that the robot has to reach. Robot motions are defined with respect to the points matched between consecutive views of the path. These sets of matches are considered as descriptions of the area the robot has to successively reach. By requiring the robot to observe these sets within good conditions, the system gets closer to the desired position. A qualitative visual servoing, using adequate objective functions, has been presented. The originality of this control law is that no particular desired positions or desired visual measures are imposed, but rather confidence intervals. Experiments realized in simulations and with a real robotic system have demonstrated the validity of the proposed approach.

Future works will consider the application of this principle to a real mobile robot. This requires us to define specific visual measures, adapted to the motions that a robotic system like a car can perform. Furthermore, we are interested in the extension of

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002

the control law in order to satisfy the non-holonomic constraints of such robotic system.

References

[1] A. Argyros, C. Bekris, S. Orphanoudakis, Robot homing based on corner tracking in a sequence of panoramic views, in: IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, USA, 2001, pp. 3–10.
[2] G. Blanc, Y. Mezouar, P. Martinet, Indoor navigation of a wheeled mobile robot along visual routes, in: IEEE Inter. Conf. on Robotics and Automation, Barcelona, Spain, 2005.
[3] D. Burschka, G.D. Hager, Vision-based control of mobile robots, in: IEEE Inter. Conf. on Robotics and Automation, Seoul, South Korea, 2001, pp. 1707–1713.
[4] D. Cobzas, H. Zhang, M. Jagersand, Image-based localization with depth-enhanced image map, in: IEEE Inter. Conf. on Robotics and Automation, Taipeh, Taiwan, 2003, pp. 1570–1575.
[5] N.X. Dao, B.J. You, S.R. Oh, M. Hwangbo, Visual self-localization for indoor mobile robots using natural lines, in: IEEE Inter. Conf. on Intelligent Robots and Systems, Las Vegas, USA, 2003, pp. 1252–1255.
[6] A. Davison, Real-time simultaneous localisation and mapping with a single camera, in: IEEE Inter. Conf. on Computer Vision, Nice, France, 2003.
[7] F. De La Torre, M.J. Black, Robust principal component analysis for computer vision, in: IEEE Inter. Conf. on Computer Vision, vol. 1, Vancouver, Canada, 2001, pp. 362–369.
[8] B. Espiau, F. Chaumette, P. Rives, A new approach to visual servoing in robotics, IEEE Transactions on Robotics and Automation 8 (3) (1992) 313–326.
[9] O. Faugeras, F. Lustman, Motion and structure from motion in a piecewise planar environment, International Journal of Pattern Recognition and Artificial Intelligence 2 (1988) 485–508.
[10] C. Harris, M. Stephens, A combined corner and edge detector, in: Alvey Vision Conf., 1988, pp. 147–151.
[11] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, England, 2000.
[12] H. Jin, P. Favaro, S. Soatto, Real-time feature tracking and outlier rejection with changes in illumination, in: IEEE Inter. Conf. on Computer Vision, vol. 1, Vancouver, Canada, 2001, pp. 684–689.
[13] S. Jones, C. Andersen, J.L. Crowley, Appearance based process for visual navigation, in: IEEE Inter. Conf. on Intelligent Robots and Systems, vol. 2, Grenoble, France, 1997, pp. 551–557.
[14] J. Košecká, L. Zhou, P. Barber, Z. Duric, Qualitative image based localization in indoor environments, in: IEEE Conf. on Computer Vision and Pattern Recognition, Madison, USA, 2003, pp. 3–10.
[15] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[16] E. Malis, F. Chaumette, 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement, International Journal of Computer Vision 37 (1) (2000) 79–97.
[17] Y. Matsumoto, M. Inaba, H. Inoue, View-based approach to robot navigation, in: IEEE Inter. Conf. on Intelligent Robots and Systems, Takamatsu, Japan, 2000, pp. 1702–1708.
[18] Y. Mezouar, A. Remazeilles, P. Gros, F. Chaumette, Image interpolation for image-based control under large displacement, in: IEEE Inter. Conf. on Robotics and Automation, vol. 3, Washington, USA, 2002, pp. 3787–3794.
[19] C. Rasmussen, G. Hager, Robot navigation using image sequences, in: Nat. Conf. on Artificial Intelligence, vol. 2, Portland, USA, 1996, pp. 938–943.
[20] A. Remazeilles, F. Chaumette, F. Gros, Image based robot navigation in 3d environments, in: Int. Symp. on Optomechatronic Technologies, ISOT'05, 2005.
[21] A. Remazeilles, F. Chaumette, P. Gros, Robot motion control from a visual memory, in: IEEE Inter. Conf. on Robotics and Automation, vol. 4, New Orleans, USA, 2004, pp. 4695–4700.
[22] E. Royer, M. Lhuiller, M. Dhome, T. Chateau, Towards an alternative gps sensor in dense urban environment from visual memory, in: British Machine Vision Conference, London, England, 2004.
[23] S. Se, D. Lowe, J. Little, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, International Journal of Robotics Research 21 (8) (2002) 735–758.
[24] A. Shashua, N. Navab, Relative affine structure: Canonical model for 3D from 2D geometry and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (9) (1996) 873–883.
[25] O. Tahri, F. Chaumette, Point-based and region-based image moments for visual servoing of planar objects, IEEE Transactions on Robotics 21 (6) (2005) 1116–1127.
[26] S. Thrun, W. Burgard, D. Fox, A real time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping, in: IEEE Inter. Conf. on Robotics and Automation, San Francisco, USA, 2000, pp. 321–328.
[27] Z. Zhang, R. Deriche, Q. Luong, O. Faugeras, A robust approach to image matching: Recovery of the epipolar geometry, in: Inter. Symposium of Young Investigators on Information–Computer-Control, 1994.
[28] C. Zhou, Y. Wei, T. Tan, Mobile robot self-localization based on global visual appearance features, in: IEEE Inter. Conf. on Robotics and Automation, Taipeh, Taiwan, 2003, pp. 1271–1276.

Anthony Remazeilles obtained an engineering degree in computer science from Institut National des Sciences Appliquées, Rennes, France, in 2001. He also obtained a M.S. Degree from the University of Rennes 1 in 2001. He received the Ph.D. degree in computer science from the University of Rennes in 2004. During the past two years, he has been a teaching assistant at INSA-IRISA, Rennes. He currently holds a postdoctoral position in CEA-LIST, Paris. His research interests include robotic and computer vision.

François Chaumette graduated from École Nationale Supérieure de Mécanique, Nantes, France, in 1987. He received the Ph.D. degree in computer science from the University of Rennes in 1990. Since 1990, he has been with IRISA/INRIA, Rennes, France, where he is now "Directeur de Recherches" and head of the Lagadic group. His research interests include robotics and computer vision, especially visual servoing and active perception.

Dr. Chaumette received the AFCET/CNRS Prize for the best French thesis in automatic control in 1991. He also received, with Ezio Malis, the 2002 King-Sun Fu Memorial Best IEEE Transactions on Robotics and Automation Paper Award. He was Associate Editor of the IEEE Transactions on Robotics from 2001 to 2005.

Please cite this article in press as: A. Remazeilles, F. Chaumette, Image-based robot navigation from an image memory, Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.10.002