3D navigation based on a visual memory

Anthony Remazeilles
INSA - IRISA
Campus de Beaulieu
35043 Rennes cedex, France
Email: Anthony.Remazeilles@irisa.fr

François Chaumette
IRISA - INRIA Rennes
Campus de de Beaulieu
35043 Rennes cedex, France
Email: Francois.Chaumette@irisa.fr

Patrick Gros
IRISA - CNRS Rennes
Campus de Beaulieu
35043 Rennes cedex, France
Email: Patrick.Gros@irisa.fr

Abstract— **This paper addresses the design of a control law for vision-based robot navigation. The method proposed is based on a topological representation of the environment. Within this context, a learning stage enables a graph to be built in which nodes represent views acquired by the camera, and edges denote the possibility for the robotic system to move from one image to an other. A path finding algorithm then gives the robot a collection of views describing the environment it has to go through in order to reach its desired position. This article focuses on the control law used for controlling the robot motion's online. The particularity of this control law is that it does not require any reconstruction of the environment, and does not force the robot to converge towards each intermediary position in the path. Landmarks matched between each consecutive views of the path are considered as successive features that the camera has to observe within its field of view. An original visual servoing control law, using specific features, ensures that the robot navigates within the visibility path. Simulation results demonstrate the validity of the proposed approach.**

I. Introduction

The possibility to use a camera for controlling the motion of a robotic system is, these days, quite a common issue. Several industrial applications benefit from this technology to automate robotic positioning tasks. Nowadays, scientific investigations concern the extension of this technology for giving to robots the capacity of moving in large environments, problem which is usually called vision-based navigation.

The difficulty of vision-based navigation comes from the fact that the initial position of the robot can be far from the position it wishes to reach. Thus, the images describing the initial and desired positions can be totally different. A robotic system able to perform such a task must therefore have an internal representation of its environment. This knowledge can then be used to localize the initial and desired positions, but also to define a path between these two positions.

In the related literature, the two major issues are the model-based and the appearance-based approaches. The first one relies on the knowledge of a $3D$ model of the navigation space. If the model is not known *a priori*, a learning step enables to reconstruct it, like in [1], where a bundle adjustment is employed to obtain the $3D$ coordinates of points, or in [2], where spherical point coordinates are deduced from the fusion of visual tracking and odometry information. A large part of the research in robot navigation is devoted to *Slam* (for *Simultaneous Localization And Mapping*), which concerns the autonomous reconstruction of the environment [3], [4], by determining the motions that enable the robot to discover new areas. But those techniques do not manage to define and realize a path for reaching a particular position.

In these approaches, the localization is nothing but a matching between the global model and the local one deduced from sensor data, such as lines [5], planes [6] or points [2], [1]. Then, the motion is usually defined as an iterative attraction towards intermediary desired positions. It can be realized either by comparing the current and the next-desired robot position [1], or by controlling that image features replay the trajectories observed during the learning step [7], [2].

Contrary to the model-based approach, the appearance-based formalism has the advantage of not requiring a $3D$ model. The scene is represented by a topological graph, where nodes correspond to local descriptions of the environment, *ie* images, and edges describe the capacity to move from one position to another. Localization is usually realized trough a similarity measure, comparing the image given by the camera and the whole data base. Several measures have been proposed, like color histograms [8], photometric invariants [9], SIFT points [10] or directly the whole image [11].

According to the information associated to the images from the database, different strategies can be employed to define the robot motions. In [11], a particular motion is associated to each image from the database, and the robot performs the movement associated to the closest view in its visual memory. This strategy requires to be sure that the robot is not going to drift from the learned trajectory, which could make the stored motions ineffective. Another common approach consists in using visual servoing schemes, based on visual landmarks extracted from the image, to converge towards each intermediary view of the image path. In [12], path planning enables to define the trajectories of points between each image. In [13], [14], set of landmarks are constrained to reach their associated positions in each image from the path. Nevertheless, it is not necessarily imperative to impose these local convergences in order to to carry out the navigation task, and it can even be problematic if the robot can not ensure it (like for example if local obstacles not present in the database make one of these convergences impossible).

This paper proposes a method related to the appearance-based formalism, which enables to get freed from an expensive and difficult global $3D$ reconstruction. The proposed method does not require to reach exactly each intermediate image, avoiding some useless local convergences. Robot motions are computed from an original adaptation of visual servoing schemes, by controlling and improving the observation of a set of interest points. Section 2 recalls how can be defined the image path. Section 3 introduces the general principle of the method proposed and gives some notations and geometrical elements that are used in the rest of the paper. Section 4 presents in details the control law proposed, and Section 5 illustrates this method with some experiences. Conclusions and further works are finally discussed in Section 6.

II. Navigation task definition within a topological approach

Figures 1 and 2 illustrate the basic operations that manage to get an image path within a topological framework. First of all, a set of images describing the environment is acquired during an off-line step. An automatic image matching algorithm (like [15]) is then used to define a relationship between each couple of images from the database. The corespondences found are used to construct the topological graph. Indeed since an edge represents a motion between two images that can perform the robot, it is necessary that the associated images describe the same part of the environment, and therefore share some points of interest.

The navigation starts with the localization of the initial robot position (see Fig. 1). In topological approaches, the localization consists in finding within the database the local sensor descriptions that are the most similar to the one given by the robotic system. In vision-based approaches, this process is usually treated as an image retrieval problem. This localization is qualitative, since the current robot position with respect to a reference frame is not searched. The retrieval process only informs that the robotic system is in the vicinity of one of the images from the database.

Once the robot has recognized its position with respect to its visual memory, the next operation consists in defining a path that links its current location with the position it has to reach (see Fig. 2). This process directly benefits from the graph representation of the database. Indeed, this path is obtained by searching in the graph the shortest path between the two nodes respectively the most similar to the current and desired positions. Since an edge links two views that share some points, the obtained path, which is called an *image path*, visually describes the environment the robotic system has to go through. If one can ensure that the robot stays during all its navigation in the vicinity of the locations described by these images, it is possible to put in relation the landmarks observed by the camera with the ones detected in the image path.

III. General approach and geometrical background

A. Navigation scheme based on feature visibility

Let $\psi_0, \ldots, \psi_N$ denotes the $N+1$ images of the path. ψ_0 is the image acquired by the camera before the robot starts to move. ψ_N is the view it wishes to obtain after the navigation. The visual memory also gives the set of correspondences between the successive images of the path. $\mathcal{M}_i$ is the set of points $\left\{ {}^{i}\mathbf{x}_j, {}^{i+1}\mathbf{x}_j \right\}$ shared by views ψ_i and ψ_{i+1}. Sets $\mathcal{M}_i$ characterize the path the robot has to go through. Indeed, at the beginning of the motion, the robot observes the landmarks of set $\mathcal{M}_0$, and it has then to move towards the area described by $\mathcal{M}_1$. Gradually, if the robot manages to observe the features of the next scheduled set $\mathcal{M}_i$, it will reach the desired area, and can then converge towards its desired position.

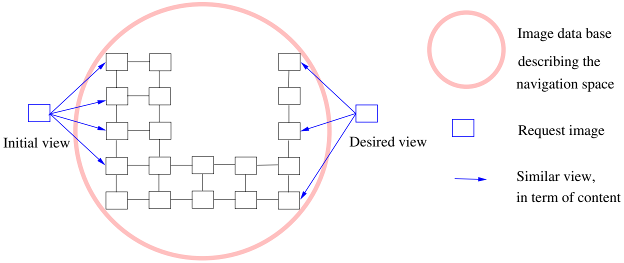


Fig. 1. Qualitative robot localization by image retrieval.

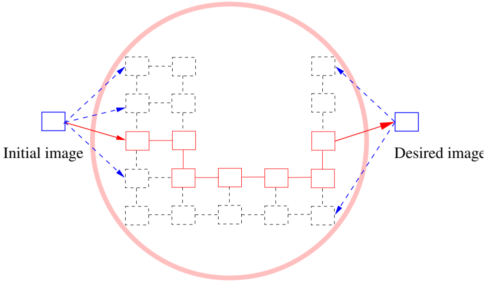


Fig. 2. Image path finding. The robot uses this path during the navigation to localize itself and to define its motion.

In order to move the robot along the path, visual features that characterize the conditions of observation of the current desired set of points $\mathcal{M}_i$ are defined. At each iteration, the motion of the robot is controlled to improve these conditions.

Figure 3 presents the control loop used during the navigation. The different steps involved are the following:

1) **Point tracking:** the points ${}^{t-1}\mathbf{x}_j$ visible in the previous view ψ_{t-1} are localized in the current view ψ_t.
2) **Point transfer:** points from the image path are transfered to the current view. It enables to determinate if new points get inside the camera field of view, but also to predict the position of points on the image plane even if they are not yet physically visible.
3) **Visible points update:** for all the set of correspondences $\mathcal{M}_i$ defined onto the image path, points that are currently projected inside the camera field of view are recorded and form the new set of visible points ${}^{t}\mathbf{x}_j$.
4) **Interest set selection:** among all the sets for which some points are already visible, the furthest one is selected, so that the robot progresses along the path.
5) **Control law update:** the motion is defined with respect to this interest set $\mathcal{M}_i$. The robot moves towards an area where the observation of this set is considered better.

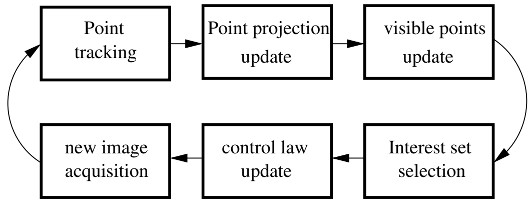


Fig. 3. General control loop used

This paper mainly focuses on the control law part, which is described in Section IV. In a real application, the tracking stage (1) can be performed with a differential point tracker [16]. Next section presents how can be performed the point transfer.

B. Point transfer with homographies

Step (2) can be realized by using the image transfer method proposed in [17], which is based on the estimation of the homography ${}^2\mathbf{H}_{p_1}$, linking the projections ${}^1\mathbf{x}_{p_j}$ and ${}^2\mathbf{x}_{p_j}$ in two views ψ_1 and ψ_2 of a $3D$ point $\mathcal{X}_j$:

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

where $\propto$ is the equality up to a scalar factor, $\mathbf{c}_2$ is the epipole of the second camera and:

$$ {}^2\mathbf{H}_{p_1} = \mathbf{K}_2 {}^2\mathbf{H}_{n_1}\mathbf{K}_1^{-1} \;\;,\; {}^2\mathbf{H}_{n_1} = \left({}^2\mathbf{R}_1 + \frac{{}^2\mathbf{t}_1 {}^1\mathbf{n}^\top}{d_\pi} \right) \tag{2}$$

$\mathbf{K}_i$ represents the camera intrinsic parameters. $({}^2\mathbf{R}_1, {}^2\mathbf{t}_1)$ is the rigid motion between the two camera positions. This rotation and translation (up to a scalar factor) can be extracted from the homography [18]. ${}^2\mathbf{H}_{p_1}$ 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 Figure 4). If all the points belong to the plane π, only four points are needed to compute the homography [18], and $\beta_{i,j} = 0$. If it is not the case, eight correspondences are needed [19].

The parallax $\beta_{1,j}$ is deduced from eq. (1):

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

By scaling the homography with respect to a point $\mathcal{X}_0 \notin \pi$, such that ${}^2\mathbf{x}_{p_0} \propto {}^2\mathbf{H}'_{p_1} {}^1\mathbf{x}_{p_0} + \mathbf{c}_2$, the parallaxes of all points become invariant to the second view [17]. 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:

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

Indeed, if one knows the homography between the current view and some images of the path, it is then possible to estimate the image coordinates of features that were not yet considered visible, even if these points are not within the camera field of view.

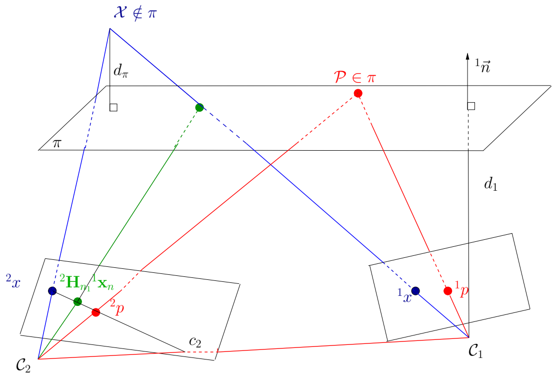


Fig. 4. Epipolar geometry deduced by the homography

Let us add that the homography enables to estimate some scene structure information [19], like the ratio between the depth Z_1 and Z_2 of a $3D$ point:

$$ \tau = \frac{Z_2}{Z_1} = \frac{\|[{}^2\mathbf{t}_1]_\times {}^2\mathbf{R}_1 {}^1\mathbf{x}_{n_j}\|}{\|[{}^2\mathbf{t}_1]_\times {}^2\mathbf{x}_{n_j}\|}, \tag{5}$$

with $\mathbf{K}_i {}^i\mathbf{x}_{n_j} = {}^i\mathbf{x}_{p_j}$, and the ratio between the depth Z_2 and distance d_1:

$$ \rho = \frac{Z_2}{d_1} = \tau \frac{\|{}^2\mathbf{t}_1/d_1\|}{\|{}^2\mathbf{t}_1/Z_1\|}, \tag{6}$$

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

IV. Qualitative Visual Servoing

The new control law we propose can be seen as a *qualitative visual servoing*. Classically, visual servoing aims at minimizing the error between the value of a set of visual features $\mathbf{s}$ and their desired value $\mathbf{s}^*$. For that, a classical control law is given by [20]:

$$ \mathbf{v} = -\lambda \mathbf{L_s}^+ (\mathbf{s} - \mathbf{s}^*), \tag{7}$$

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 $\mathbf{s}$ (defined such as $\dot{\mathbf{s}} = \mathbf{L_s v}$).

In the method proposed, no particular desired position can be defined, since the robot is not required to reach each intermediary position defined by the image path. As we previously mentioned, the robot is only imposed to move along areas where the conditions of observation of the set $\mathcal{M}_i$ are considered satisfactory. This observation quality is controlled by a set of objective functions $\mathcal{V}$ defined such that their minima correspond to positions where the associated criterion of observation is satisfied.

Therefore, the error $\mathbf{s} - \mathbf{s}^*$ in (7) is replaced by the gradient of $\mathcal{V}(\mathbf{p})$. A satisfactory area is reached when $\nabla\mathcal{V}(\mathbf{p}) = 0$. In other words, $\mathbf{s} = \nabla\mathcal{V}(\mathbf{p})$, $\mathbf{s}^* = \nabla\mathcal{V}(\mathbf{p})^* = \mathbf{0}$, and $\mathbf{L_s}$ is replaced by the interaction matrix of $\nabla\mathcal{V}(\mathbf{p})$.

Following subsections present the different functions proposed to progress along the path.

A. Progressing along the path

This first objective function is designed to move the robot forward. The projection on the image plane of a set of $3D$ points is closely related to the distance between the points and the camera. It is obvious that if this distance is increased, the projections get closer to the image center (since $x = X/Z$). Thus, the comparison of the projection areas in the current view and in the next image of the path gives a good estimate of the distance between the two camera positions.

This projection area can be described by the following measurement, which is based on second order centered moments:

$$a = \mu_{02} + \mu_{20}, \text{ with } \mu_{ij} = \sum_{k=0}^{n} \left(x_k - x_g\right)^i \left(y_k - y_g\right)^j, \quad (8)$$

where (x_g, y_g) is the center of gravity of the n considered points. The following measure a_n enables then to compare the current area with the one observed in the image ψ_{i+1}:

$$a_n = \sqrt{\frac{a^*}{a}}, \quad (9)$$

Since the robot is not required to reach the next image of the path ψ_{i+1}, we are not expecting to obtain the same measure a_n^*, but rather a value in the vicinity:

$$a_m = a_n^*(1-p) < a_n < a_n^*(1+p) = a_M, \quad (10)$$

with $p \in [0\ 1]$. The validity of this inequality can be controlled with the following function:

$$\mathcal{V}_{a_n}(a_n) = g(a_n - a_M) + g(a_m - a_n), \quad (11)$$

where:

$$g(x) = \frac{1}{2}x^2 h(x) \text{ and } h(x) = \frac{\arctan(k\pi x)}{\pi} + \frac{1}{2} \quad (12)$$

$h(x)$ is the arc-tangent function normalized on $[0;1]$. It corresponds to an *"heavy-side"* function that defines a smooth transition between values 0 and 1, starting when $x = 0$. The constant scalar k regulates the curvature of the transition. $\mathcal{V}_{a_n}$ is thus null when a_n belongs to the confident interval and tends towards the parabolic function when a_n moves away from this free area (see Fig. 5).

The error associated to $\mathcal{V}_{a_n}$ is defined as:

$$e_{a_n} = \nabla_{a_n}\mathcal{V}_{a_n} = \frac{\partial \mathcal{V}_{a_n}}{\partial a_n}, \quad (13)$$

where $\nabla_{a_n}\mathcal{V}_{a_n}$ is easily obtained from (11). The derivative of e_{a_n} with respect to time is:

$$\dot{e}_{a_n} = \frac{\partial e_{a_n}}{\partial a_n}\frac{da_n}{dt} = \frac{\partial^2 \mathcal{V}_{a_n}}{\partial a_n^2}\mathbf{L}_{a_n}\mathbf{v} = \mathbf{L}_{e_{a_n}}\mathbf{v}, \quad (14)$$

where $\mathbf{L}_{a_n}$ is the interaction matrix related to a_n, and $\mathbf{L}_{e_{a_n}}$ the one related to the visual feature $\nabla_{a_n}\mathcal{V}_{a_n}$. This matrix must be estimated in order to define the control law.

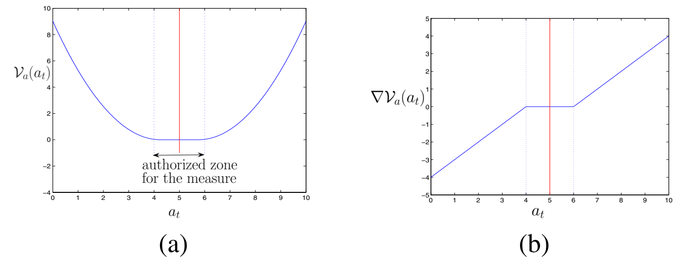


Fig. 5. Controlling motions along the optical axis: used, (b) its gradient

As previously mentioned, the function h used in g realizes a smooth transition between values 0 and 1. The objective function defined in (11) can thus be approximated by :

$$\mathcal{V}_{a_n}(a_n) = \begin{cases} \frac{1}{2}\left(a_m - a_n\right)^2 & \text{if } a_n < a_m, \\ \frac{1}{2}\left(a_n - a_M\right)^2 & \text{if } a_n > a_M, \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

We have of course:

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

Since $e_{a_n} = 0$ for $a_m < a_n < a_M$, $\mathbf{L}_{a_n}$ can be chosen as a good approximation of $\mathbf{L}_{e_{a_n}}$. $\mathbf{L}_{a_n}$ is given by [21]:

$$\mathbf{L}_{a_n} = [0\ 0\ -1/Z^*\ -a_n\epsilon_1\ a_n\epsilon_2\ 0], \quad (17)$$

with:

$$\begin{array}{rl} \epsilon_1 = & y_g + \left(y_g\mu_{02} + x_g\mu_{11} + \mu_{21} + \mu_{03}\right)/a \\ \epsilon_2 = & x_g + \left(x_g\mu_{20} + y_g\mu_{11} + \mu_{12} + \mu_{30}\right)/a, \end{array} \quad (18)$$

where ϵ_1 and ϵ_2 can be neglected with respect to 1. Eq. (17) is correct only if the camera is parallel to a supposed planar object (at a distance Z^*). However this matrix is used in our application (with Z^*=1), since we do not require to get the same measure. Indeed, we do not impose to reach exactly the next image. This approximation is thus enough to move the robot closer to the position related to the next view of the path.

B. Feature position control

The next function controls the measure projection onto the image plane. All the elements of the measure set $\mathcal{M}_i$ should project within the camera field of view. Therefore, a free area $\mathcal{I}_{free}$ is defined within the image plane (see Fig. 6). A point projection $\mathbf{x}_j = (x_j, y_j)$ is considered satisfactory if it is 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 $\alpha > 0$.

The function $\mathcal{V}_{\mathbf{s}}$ that enables to characterize the quality of the point positions on the image plane is defined by:

$$\mathcal{V}_{\mathbf{s}} = \sum_j \mathcal{V}_{s_{(\mathbf{x}_j)}} \quad (19)$$

with:

$$\mathcal{V}_{s_{(\mathbf{x}_j)}} = g(x_m - x_j) + g(x_j - x_M) + g(y_m - y_j) + g(y_j - y_M), \quad (20)$$

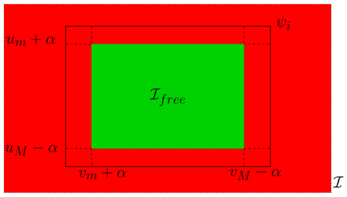


Fig. 6. $\mathcal{I}_{free}$ is a restriction of the infinite image plane $\mathcal{I}$

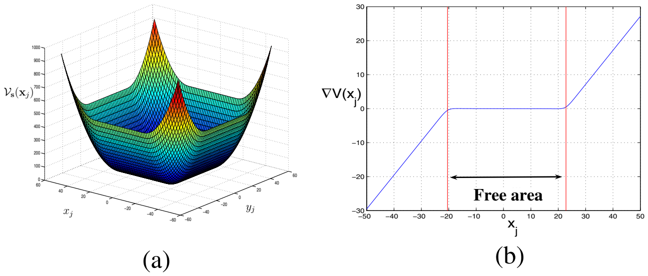


Fig. 7. Controlling point positions: (a) function for a single point, (b) gradient for x_j coordinates

where $g(x)$ has already been given in equation (12). Figure 7 presents this objective function and its gradient.

By using the same approximation as before, the interaction matrix related to $\nabla_{\mathbf{s}}\mathcal{V}_{\mathbf{s}}$ is approximated by the interaction matrix $\mathbf{L}_{\mathbf{s}}$ associated to the image point coordinates. This matrix stacks the different interaction matrices of all considered points. For a single point, this matrix is:

$$\mathbf{L}_{\mathbf{x}_j} = \left[\begin{array}{cc} \frac{1}{d_{i+1}}\mathbf{S}_j & \mathbf{Q}_j \end{array} \right], \tag{21}$$

with:

$$\mathbf{S}_j = \frac{1}{\rho_j}\left[\begin{array}{ccc} -1 & 0 & x_j \\ 0 & -1 & y_j \end{array} \right] \; \mathbf{Q}_j = \left[\begin{array}{ccc} x_j y_j & -(1+x_j^2) & y_j \\ 1+y_j^2 & -x_j y_j & -x_j \end{array} \right] \tag{22}$$

d_{i+1} is the distance between the camera frame of view ψ_{i+1} and the reference plane π. Scalar ρ_j can be extracted from the homography ${}^t\mathbf{H}_{n_{i+1}}$ between the current view ψ_t and the image ψ_{i+1} of the path (see eq. (6)).

C. Control of the orientation

The last function deals with the error of orientation that can be found between the current camera position and the next image ψ_{i+1} of the path. This rotation ${}^t\mathbf{R}_{i+1}$ can be extracted from the homography ${}^t\mathbf{H}_{i+1}$ estimated from matches obtained between the current view and the view ψ_{i+1}.

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

$$\theta\mathbf{u} = \frac{1}{2\text{sinc}\theta}\left(\begin{array}{c} r_{32} - r_{23} \\ r_{31} - r_{13} \\ r_{21} - r_{12} \end{array} \right), \tag{23}$$

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

Once more, the quality of the current orientation is defined by an interval:

$$-p_\theta < \theta u_i < p_\theta, \tag{24}$$

where $p \in [0\ 1]$. The associated function is:

$$\mathcal{V}_{\theta\mathbf{u}}(\theta u_i) = g(\theta u_i - p_\theta) + g(-p_\theta - \theta u_i), \tag{25}$$

and the interaction matrix of $\nabla_{\theta\mathbf{u}}\mathcal{V}_{\theta\mathbf{u}}$ is approximated by $\mathbf{L}_{\theta\mathbf{u}}$ [19]:

$$\mathbf{L}_{\theta\mathbf{u}} = [\mathbf{0}_3\ \mathbf{L}_w] \text{ where } \mathbf{L}_w = \mathbb{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\text{sinc}\theta}{\text{sinc}^2\frac{\theta}{2}}\right)[\mathbf{u}]_\times^2 \tag{26}$$

Let us notice that in our experiments, this function is only used to control rotations around $\vec{x}$ and $\vec{y}$ axis. Indeed, rotations around $\vec{z}$ do not either improve the feature visibility or move the robot towards its goal. But it could be possible to control this degree of freedom for other applications.

Next subsection presents how these different functions are combined to compute the motion of the robotic system.

D. General control law

In order to simultaneously satisfy the three constraints previously defined, the control law is given by:

$$\mathbf{v} = -\lambda\mathbf{L}^{-1}\nabla^\top, \tag{27}$$

where $\mathbf{L}$ and ∇ are respectively a stack of interaction matrices and visual features:

$$\nabla = \left(\nabla_{a_n}\mathcal{V}_{a_n}, \nabla_{\mathbf{s}}^\top\mathcal{V}_{\mathbf{s}}, \nabla_{\theta\mathbf{u}}^\top\mathcal{V}_{\theta\mathbf{u}}\right) \text{ and } \mathbf{L} = (\mathbf{L}_{a_n}, \mathbf{L}_{\mathbf{s}}, \mathbf{L}_{\theta\mathbf{u}}) \tag{28}$$

The feature projection ${}^t\mathbf{x}_{n_j}$ of points belonging to $\mathcal{M}_i$ (shared by views ψ_i and ψ_{i+1}), and the homography ${}^t\mathbf{H}_{i+1}$ estimated from correspondences between ψ_{i+1} and ψ_t are used to update ∇ and $\mathbf{L}$, as proposed in the previous subsections.

Finally, once enough features originally matched between views ψ_{N-1} and ψ_N are visible, a classical visual servo control law is used to converge towards the desired position.

V. Experimental Results

This section presents several experiments that validate the navigation scheme proposed. These experiments are performed in simulation, in order to study the control law behavior without adding potential noise that could bring the tracking and prediction steps.

A. Five degrees of freedom system

Figure 8 presents the environment in which the camera is moving. It is composed of a set of planes onto which are defined some points of interest.

Figure 9 describes the camera poses associated to the different views of the path, that are contained in the image data base. Figure 10(a) displays the initial and desired view, and Figure 10(b) presents, as an example, the features matched between views ψ_1 and ψ_2 of this path.

The trajectory and the camera poses obtained with the proposed control law are presented on Figures 11 and 12. In Figure 12, blue vertical lines denotes a change of interest set $\mathcal{M}_i$, and crosses indicates the position of the reference frame used just before this change. Even if the obtained trajectory implicitely depends on the image path, the displacement

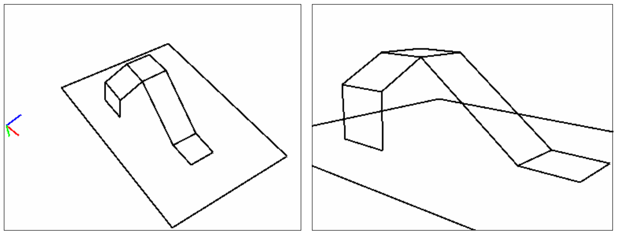


Fig. 8. Views of the 3D object used for simulations. The camera frame is represented as follows: $\vec{x}$ axis in red, $\vec{y}$ axis in green, and $\vec{z}$ axis in blue

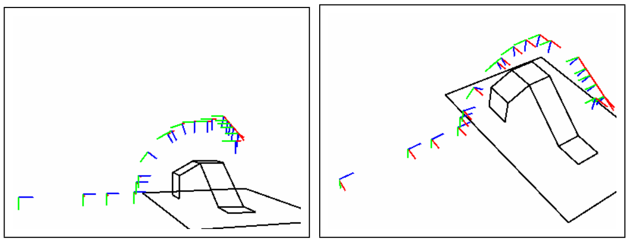
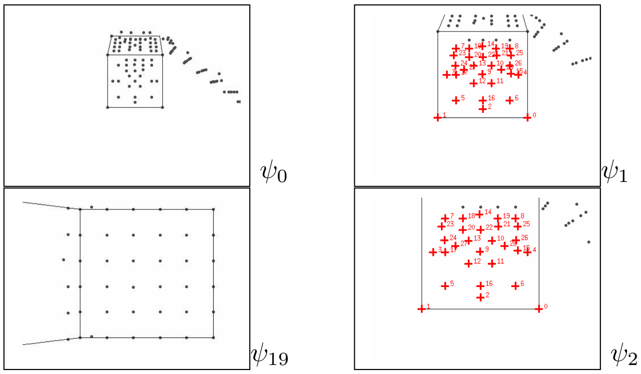


Fig. 9. Exp. 1: Poses corresponding to the image path



(a) (b)

Fig. 10. Exp. 1: Illustration of the image path (a) initial and desired view (b) example of a matched set ($\mathcal{M}_1$, between views ψ_1 and ψ_2)

realized is not a succession of convergences towards each intermediary views. As scheduled, the beginning of the motion is mainly a translation along the optical axis (iterations 0 to 370). Once the camera is close enough to the object, translations along $\vec{y}$ axis and rotations around $\vec{x}$ axis are realized in order to observe the features suited on the upper part of the object (370 to 800). Translations along $\vec{x}$ axis permit then to reach the desired final area. Finally, a classical positioning control law is used for performing the final convergence (which is not shown on Figure 12).

In the next experience, the image path used is the same, but the initial position has changed (see Fig. 13). As it can be seen on Figure 15, the overall motion is very similar to the previous one. Nevertheless, one can observe that at the beginning of the motion, translations along $\vec{x}$ axis and rotations around $\vec{y}$ axis are larger (from iteration 0 to 370). Indeed, these motions enable to keep the object within the camera field of view, since translations along the optical axis make the object projection get closer to the left border of the image.

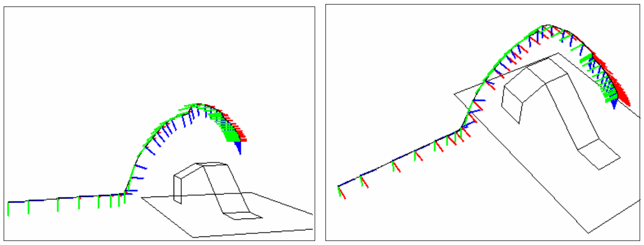


Fig. 11. Exp.1: realized trajectory

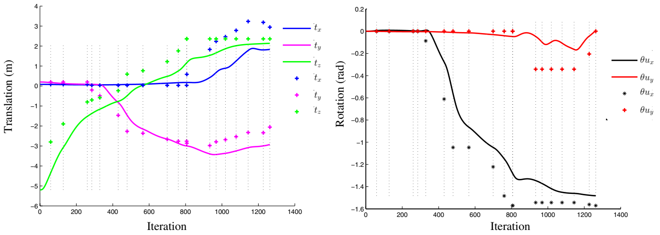


Fig. 12. Exp.1: Position (Translation et Orientation) of the camera during the navigation. Crosses indicate the position of the image used for controlling the motion. Vertical lines denote a change of interest set.

B. Set up two: robotic system moving on a plane

In the next experience, the proposed control law is used for controlling the motion of a camera moving on a plane, as if it was mounted on a holonomic mobile robot. The navigation space corresponds to a corridor, defined by a set of planes on the floor and on the walls.

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

Figure 16(a) represents the image path used here. One can notice that some images of the path are not aligned with the direction the robot should follow. Figure 16(b), presents the trajectory realized during the navigation. The system does not perform the useless convergences towards images ψ_1 and ψ_3 of the path, and manages to reach the desired position. Figure 17 gives the position of the robotic system during the motion. The change of interest feature set is always performed before the convergence.

VI. Conclusion

This article has presented a new control law for robot navigation. It is assumed that an image path can be extracted from a visual memory describing the environment. This image path defines the visual features that the camera should observed during the motion. The control law proposed does not require a $3D$ reconstruction of the environment. Furthermore, images of 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 area the robot has to successively reach. By requiring the robot to observe these sets within good conditions, the system

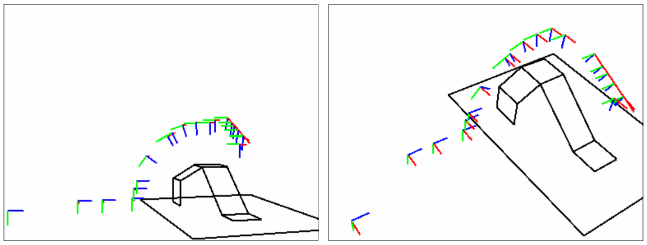


Fig. 13. Exp.2: image path positions

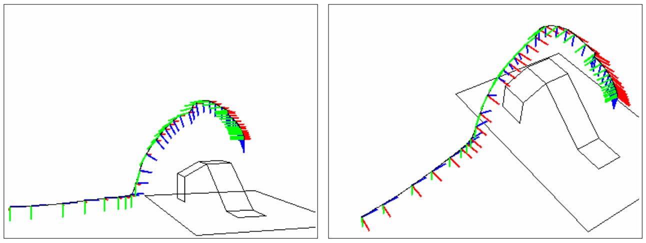


Fig. 14. Exp. 2: realized trajectory

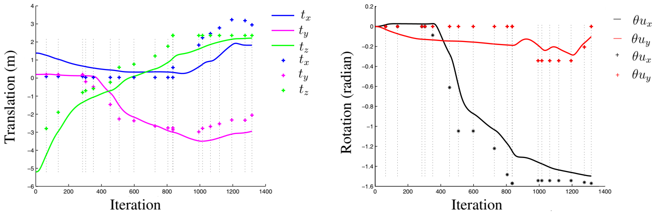


Fig. 15. Exp. 2: Position of the camera during the navigation.

gets closer to the desired position. A qualitative visual servoing control law has been presented. The originality of this control law is that no particular desired positions or desired visual measures are imposed, but rather an interval of confidence. Experiments realized in simulations have shown the validity of the proposed approach.

Further works will consider the application of this principle to a real mobile robot. This requires to define specific visual measures, adapted to the motion that can perform a robotic system like a car. Furthermore, we are interested in the extension of the control law in order to satisfy the non-holonomic constraints of such robotic system.

REFERENCES

[1] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau, “Towards an alternative gps sensor in dense urban environment from visual memory,” in *British Machine Vision Conference*, London, England, Sept. 2004.

[2] D. Burschka and G. D. Hager, “Vision-based control of mobile robots,” in IEEE *ICRA*, Seoul, South Korea, May 2001, pp. 1707–1713.

[3] S. Thrun, W. Burgard, and D. Fox, “A real time algorithm for mobile robot mapping with applications to multi-robot and 3*d* mapping,” in IEEE *ICRA*, San Francisco, USA, April 2000, pp. 321–328.

[4] S. Se, D. Lowe, and J. Little, “Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks,” *Intern. Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[5] N. X. Dao, B. J. You, S. R. Oh, and M. Hwangbo, “Visual self-localization for indoor mobile robots using natural lines,” in IEEE *IROS*, Las Vegas, USA, 2003, pp. 1252–1255.

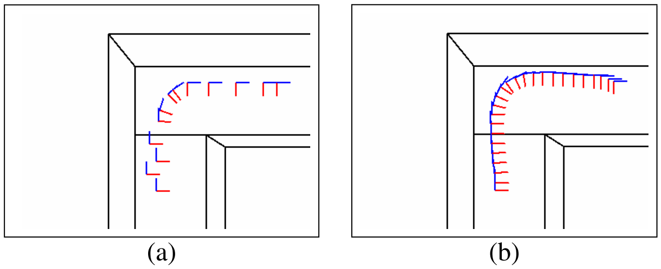


Fig. 16. Exp. 3: (a) image path positions and (b) realized trajectory

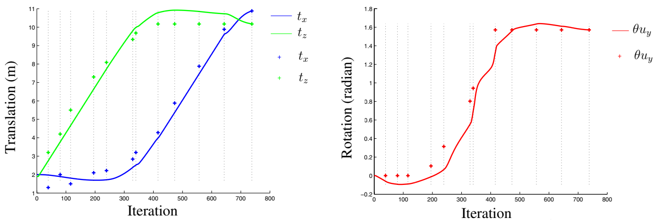


Fig. 17. Exp. 3: Position of the camera during the navigation.

[6] D. Cobzas, H. Zhang, and M. Jagersand, “Image-based localization with depth-enhanced image map,” in IEEE *ICRA*, Taipeh, Taiwan, May 2003, pp. 1570–1575.

[7] C. Rasmussen and G. Hager, “Robot navigation using image sequences,” in *National Conference on Artificial Intelligence*, vol. 2, Portland, USA, 1996, pp. 938–943.

[8] J. Košecka, L. Zhou, P. Barber, and Z. Duric, “Qualitative image based localization in indoor environments,” in IEEE *CVPR*, Madison, USA, June 2003, pp. 3–10.

[9] A. Remazeilles, F. Chaumette, and P. Gros, “Robot motion control from a visual memory,” in IEEE *ICRA*, vol. 4, New Orleans, USA, April 2004, pp. 4695–4700.

[10] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Intern. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] Y. Matsumoto, M. Inaba, and H. Inoue, “View-based approach to robot navigation,” in IEEE *IROS*, Takamatsu, Japan, Oct. 2000, pp. 1702–1708.

[12] Y. Mezouar and F. Chaumette, “Path planning for robust image-based control,” IEEE *Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, August 2002.

[13] A. Argyros, C. Bekris, S. Orphanoudakis, and L. Kavraki, “Robot homing by exploiting panoramic vision,” *Journal of Autonomous Robots*, vol. 19, no. 1, pp. 7–25, July 2005.

[14] G. Blanc, Y. Mezouar, and P. Martinet, “Indoor navigation of a wheeled mobile robot along visual routes,” in IEEE *ICRA*, Barcelona, Spain, April 2005.

[15] Z. Zhang, R. Deriche, Q. Luong, and O. Faugeras, “A robust approach to image matching : Recovery of the epipolar geometry,” *Intern. Symposium of Young Investigators on Information-Computer-Control*, 1994.

[16] H. Jin, P. Favaro, and S. Soatto, “Real-time feature tracking and outlier rejection with changes in illumination,” in IEEE *Intern. Conference on Computer Vision*, vol. 1, Vancouver, Canada, July 2001, pp. 684–689.

[17] A. Shashua and N. Navab, “Relative affine structure: Canonical model for 3*d* from 2*d* geometry and applications,” IEEE *Trans. on PAMI*, vol. 18, no. 9, pp. 873–883, Sept. 1996.

[18] O. Faugeras and F. Lustman, “Motion and structure from motion in a piecewise planar environment,” *Intern. Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, pp. 485–508, 1988.

[19] E. Malis, F. Chaumette, and S. Boudet, “2 1/2 d visual servoing,” IEEE *Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, April 1999.

[20] C. Samson, M. Le Borgne, and B. Espiau, *Robot control: The Task function approach*. Oxford: Clarendon Press, 1991.

[21] O. Tahri and F. Chaumette, “Image moments: Generic descriptors for decoupled image-based visual servo,” in IEEE *ICRA*, vol. 2, New Orleans, USA, April 2004, pp. 1185–1190.