Image-based robot navigation from an image memory

A. Remazeilles, François Chaumette

Image-based robot navigation from an image memory

Anthony Remazeilles, François Chaumette

IRISA,
Campus de Beaulieu
35042 Rennes Cedex, France
FirstName.LastName@irisa.fr

Abstract

This paper addresses the problem of vision-based navigation and proposes an original control law to perform the 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 space. Within this context, a path to perform is described by a set of images, or *image path* extracted from the database. This image path is designed so that it provides enough information to control the robotic system. The central point 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 neither a global $3D$ reconstruction, nor a temporal planning step. Furthermore, the robot is not constrqined to converge directly upon each image of the path but chooses automatically its trajectory. We propose a *qualitative visual servoing*, enabling to enlarge the convergence space towards an interval of confident position. 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). Furthermore, experiments realized with a robotic arm observing a planar scene are also presented.

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 from each other. 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 with the sensor information the robot should obtain at the desired position.

It is thus obvious that the robotic system needs a representation of its environment for performing 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 the local model deduced from sensor data. Features used can be either lines [Dao et al., 2003], planes [Cobzas et al., 2003] or points [Burschka and Hager, 2001, Royer et al., 2004]. 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 [Royer et al., 2004], where the reconstruction is performed using a hierarchical bundle adjustment, or like in [Burschka and Hager, 2001] where odometry is coupled

with a visual tracking system for estimating spherical feature coordinates. A large amount of articles proposes an autonomous mapping of the environment (methods known as SLAM, for *Simultaneous Localization And Mapping*) [Thrun et al., 2000, Se et al., 2002, Davison, 2003]. In this case, autonomous motions are performed for discovering new areas, but not for reaching a particular desired position.

Once the current robot position is estimated, its motion is generally performed by attracting it towards intermediary desired positions. In [Royer et al., 2004], motion is deduced from the error measured between the current robot position and the one associated to an intermediary view. In [Rasmussen and Hager, 1996, Burschka and Hager, 2001], the motion is obtained by imposing the features to follow 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 the $3D$ model of the environment. It presents the advantage to work 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 [Jones et al., 1997, Matsumoto et al., 2000], color histograms [Zhou et al., 2003], or image gradient [De La Torre and Black, 2001, Košecká et al., 2003]). Another method consists of taking advantage of image retrieval principles to localize the robot, by using local descriptors, like photometric invariants [Remazeilles et al., 2004] or SIFT points [Lowe, 2004].

Different strategies are then proposed to control the robot during the navigation. In [Jones et al., 1997, Matsumoto et al., 2000], a particular motion is associated to each image of the database. At each iteration, the robot performs the motion associated to the closest view of the sequence. However this scheme can not take into account a potential deviation from the pre-taught path, which can be problematic. In [Argyros et al., 2001, Blanc et al., 2005], 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 are initial and desired positions. Furthermore, it can be considered as useless to converge towards each intermediary positions, as long as these local convergences are not necessary for 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 merging all the information in a common $3D$ frame.

Figures 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 how the localization is performed, before the beginning of the motion. During this step, no assumption is made about the robotic system position. One can notice that it is not a particular hypothesis, which can also be found in a large set of works on localization [Košecká et al., 2003, Zhou et al., 2003]. The only information used corresponds to the set of images acquired during an off-line step (this database is surrounded by an orange circle on the figure). The localization consists in finding the views of the database that are similar, in term of content, to the request images, either the initial one or the desired one. On the figure, blue arrows describes the similar views that are found. In [Remazeilles et al., 2005], 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, but only the most similar views.

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 figure 2. This subset of views is directly deduced from the structure of the database. Indeed, like every topological approaches, the different views describing the environment are organized within a graph. Each node represents an image, and an edge between two nodes defines that the robot can move autonomously between the two associated views. The similar images obtained in the previou step enable to incorporate the initial and desired images to 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 figure 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 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 such that the visual features used for controlling the system are regulated toward confident 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 5 contains concluding remarks.

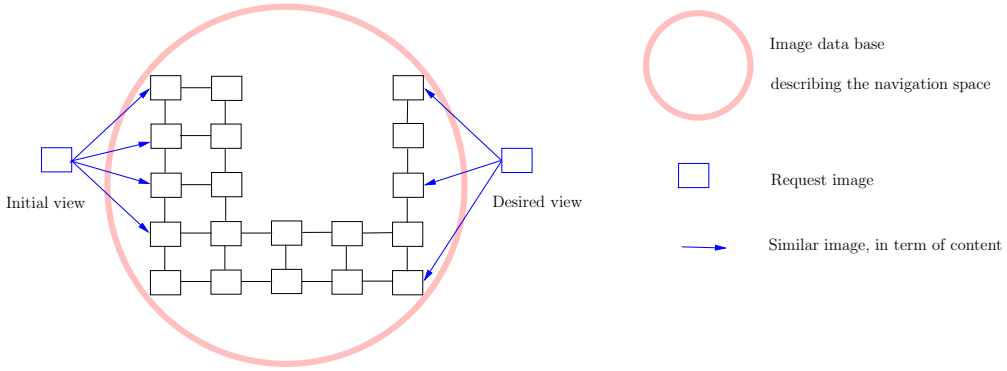


Figure 1: Qualitative localization of the robotic system by image retrieval.

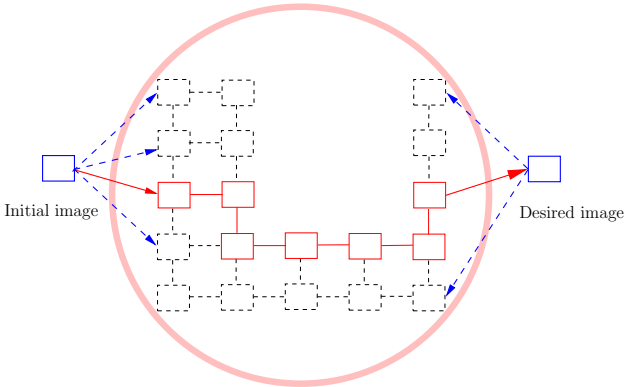


Figure 2: Image path finding. The robot will use this path to control its motion during the navigation.

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 should acquire the camera when the robot reaches its desired position. Features used during the navigation are Harris points [Harris and Stephens, 1988] matched between the consecutive images of the path. Methods like [Zhang et al., 1994] can be used to determine these correspondances. 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. These features describe therefore 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 projection of this feature set, the motion of the robotic system aims to make these visual measures reach confident intervals so that the robot becomes enable to observe the next set of points $\mathcal{M}_{i+1}$.

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

Figure 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 to determine if new features get inside the camera field of view.
3. **Visible points update:** for all the set of correspondences $\mathcal{M}_i$ defined onto the image path, features 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. It describes all the features the camera should observe.
5. **Control law update:** considering the interest set, the motion of the robot is computed. This motion enables to move the robot towards an area in which the visibility of the whole set is considered as better.

The tracking stage (1) can be performed in a real application with a differential point tracker like [Jin et al., 2001]. 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 ${}^2\mathbf{H}_{p_1}$, trough the equation [Hartley and Zisserman, 2000]:

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

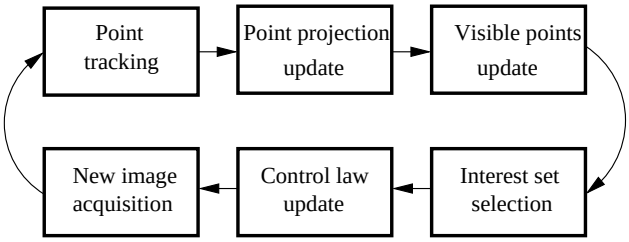


Figure 3: General control loop used

with :

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

$\mathbf{K}$ represents the camera intrinsic parameters, and 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 translation (up to a scalar factor) can be extracted from the homography [Faugeras and Lustman, 1988]. 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 Figure 4).

If all the points observed belong to the reference plane, only four points are needed for computing the homography [Faugeras and Lustman, 1988], and $\beta_{i,j} = 0$. If it is not the case, eight correspondences are needed [Malis and Chaumette, 2000].

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 notice in eq. (2) than the parallax term is independent to the second frame position. Nevertheless, as the epipole is only known up to a scalar factor, the equation (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 $\mathcal{X}_0 \notin \pi$ [Shashua and Navab, 1996]:

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

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 to determine some scene structure information, like the ratio between the depth Z_1 and Z_2 of a $3D$ point [Malis and Chaumette, 2000]:

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

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.

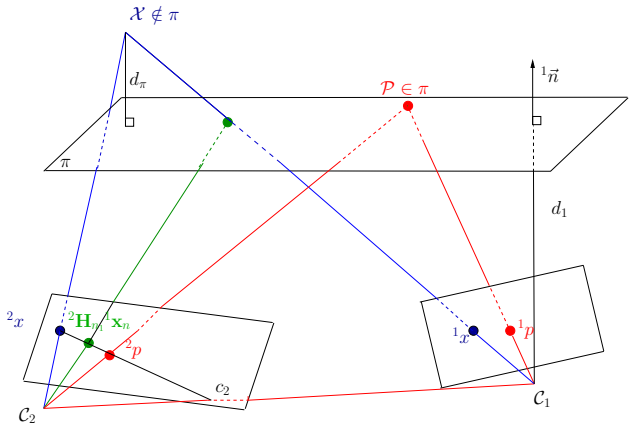


Figure 4: Relation induced by an homography between two views

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 [Espiau et al., 1992]:

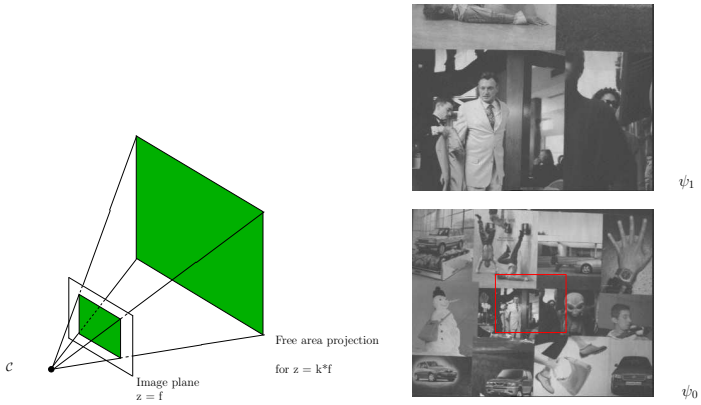
$$\mathbf{v} = -\lambda \mathbf{L}_\mathbf{s}^+ (\mathbf{s} - \mathbf{s}^*), \tag{6}$$

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 in 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 minimum correspond to poses where the associated visual feature belongs to $\mathbf{s} \in [\mathbf{s}_{min}; \mathbf{s}_{max}]$. Then, the gradient 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:

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



(a) (b)

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

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

Subsections 2.3.2, 2.3.3 and 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. Subsection 2.3.5 finally describes how they are merged together.

2.3.2 Progressing along the path

This first objective function is dealing with the camera motions along the optical axis. When considering a pinhole camera, as illustrated on Figure 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 is Z, the closer to the image center is the point projection. 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 translation t_z, then the projection of a point becomes in the second view $(X/(t_z + Z), Y/(t_z + Z))$. The further the current camera is to the next image, the closer to the image center is the point projection. This point is illustrated on Figure 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 exactly, 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 are the points to the camera, the bigger is the value of a. 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 [Tahri and Chaumette, 2005]:

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

Since the robot is not required to reach exactly each position associated to the image path, it is not imposed 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 such that:

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

This could be controlled with the following function:

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

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

The error associated to $\mathcal{V}_{a_n}$ 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 function h and $\mathcal{O}$ used for continuity matters, this gradient can be approximated by:

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

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

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

where $\mathbf{L}_{a_n}$ is the interaction matrix related to a_n and $\mathbf{L}_{\nabla_{a_n}}$ the one associated to the visual feature $\nabla_{a_n}\mathcal{V}_{a_n}$. By using the approximation proposed on eq. (14), we get:

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

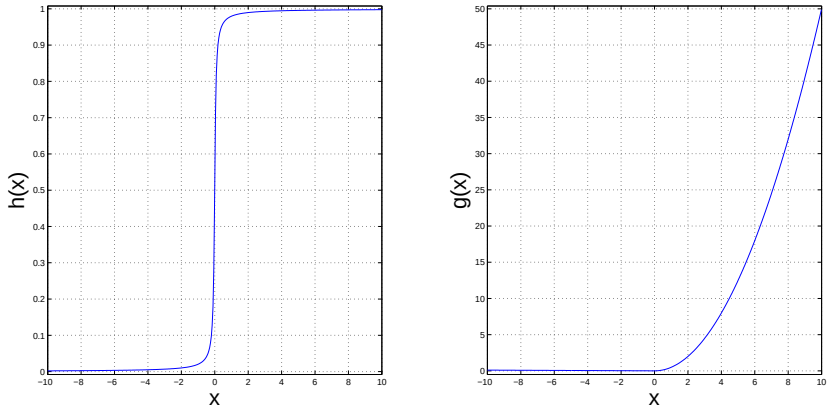
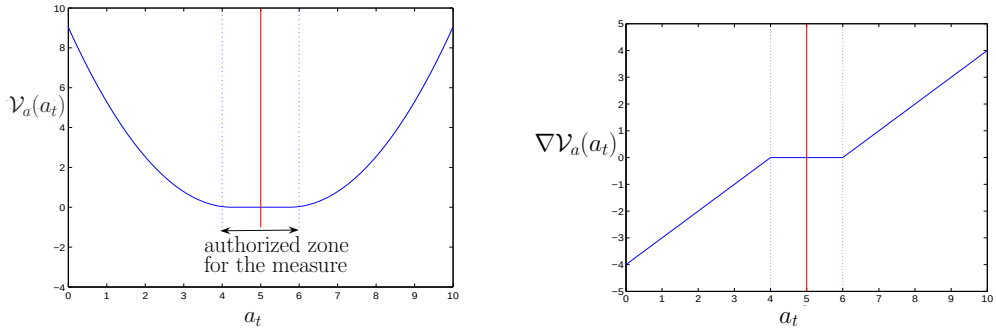


Figure 6: Functions h and g used to smooth the objective function (see eq. (11)).



(a) (b)

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

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, 2005]:

$$\mathbf{L}_{a_n} = \left[0\ 0\ -\frac{1}{Z^*}\ -a_n\epsilon_1\ a_n\epsilon_2\ 0\right], \tag{17}$$

with:

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

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 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 sounds 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 a positive constant defining a free projection area $\mathcal{I}_{free}$ within the image frame (see Figure 8).

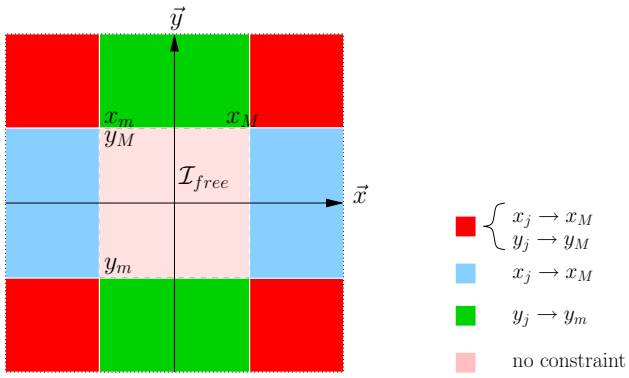


Figure 8: Areas of activation induced in the control law, displayed on the image plane ($\mathcal{I}_{free}$ is a restriction of the image frame).

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

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

where $g(x)$ has already been given in eq. (11). Figure 9 represents this objective function for a single point, and one component of its gradient. In the application considered here, $\nabla_\mathbf{s}^\top \mathcal{V}_\mathbf{s}$ gathers the gradients of the different features of the interest set $\mathcal{M}_i$:

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

where:

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

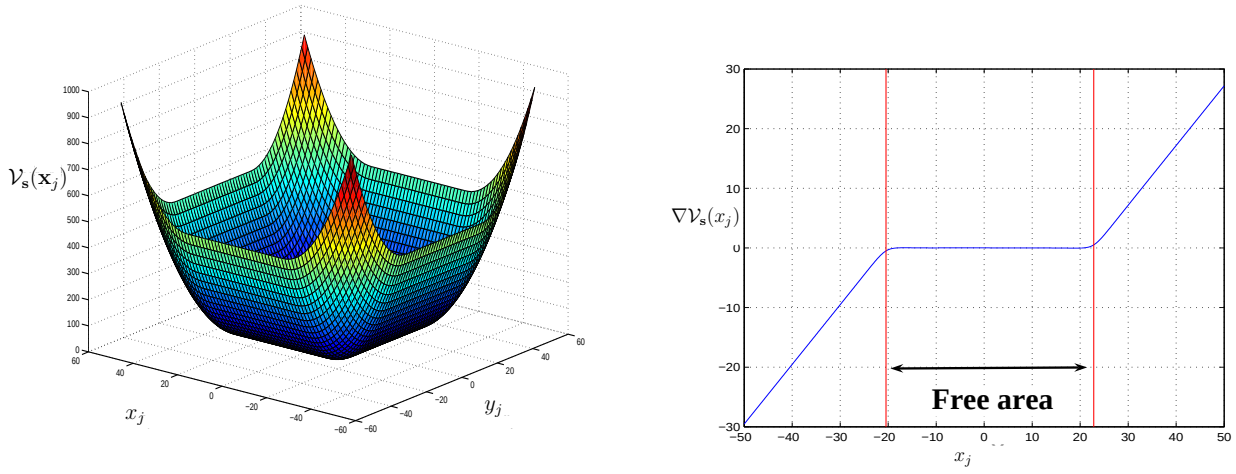
By using the same approximation as before, the interaction matrix related to $\nabla_\mathbf{s}^\top \mathcal{V}_\mathbf{s}$ is approximated by the interaction matrix $\mathbf{L}_\mathbf{s}$ associated to the image point coordinates. This matrix is given by:

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

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}:

$$\mathbf{S}_j = \frac{1}{\rho_j} \begin{bmatrix} -1 & 0 & x_j \\ 0 & -1 & y_j \end{bmatrix} \quad \mathbf{Q}_j = \begin{bmatrix} x_j y_j & -(1 + x_j^2) & y_j \\ 1 + y_j^2 & -x_j y_j & -x_j \end{bmatrix}$$

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



(a) (b)

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

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 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:

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

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

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:

$$\begin{array}{rcl} \nabla_{\theta\mathbf{u}}\mathcal{V}_{\theta\mathbf{u}}(\theta u_i) & = & (\theta u_i - p_\theta)h(\theta u_i - p_\theta) + \mathcal{O}(\theta u_i - p_\theta) \\ & + & (\theta u_i + p_\theta)h(-p_\theta - \theta u_i) - \mathcal{O}(\theta u_i + p_\theta) \end{array} \tag{19}$$

The interaction matrix of $\nabla^\top_{\theta\mathbf{u}}\mathcal{V}_{\theta\mathbf{u}}$ is approximated by $\mathbf{L}_{\theta\mathbf{u}}$ [Malis and Chaumette, 2000]:

$$\mathbf{L}_{\theta\mathbf{u}} = [\mathbf{0}_3 \;\; \mathbf{L}_w], \quad \text{where} \quad \mathbf{L}_w = \mathbb{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\text{sinc}\theta}{\text{sinc}^2\frac{\theta}{2}}\right)[\mathbf{u}]^2_\times \tag{20}$$

The function defined here is very similar to the ones defined before, corresponding curves are therefore not shown. Let us notice 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 either the feature visibility or 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 theses 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 describing each a different constraint on the visual feature configurations, and all these constraints have to be respected simultaneously. It 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:

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

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 views ψ_i and ψ_{i+1} of the path. Their coordinates are obtained either by the tracking step, either 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 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 classical visual servoing scheme can then be used to converge towards this position.

To conclude, the control law proposed can be considered as a *qualitative visual servoing*. The word *qualitative* means that there is not a single position that enables to have this convergence. Indeed, contrary to classical visual servoing, no particular desired value is required, but an interval of confident ones.

Furthermore, one can notice 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 made the success of the 2 $1/2D$ visual servoing proposed in [Malis and Chaumette, 2000].

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 five degrees of freedom camera, and the next subsection for a camera moving on a plane as if it was mounted on a holonomic mobile robot. Finally, subsection 3.2 presents some experiments realized on a real robotic system, with a planar environment.

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 Figure 10). At each face is associated a set of points. All the points defined do not belong to a particular face.

On Figure 11, a navigation task is presented by the set of positions associated to the image path that is contained in the image database. Some of the corresponding views are presented on Figure 12. The position

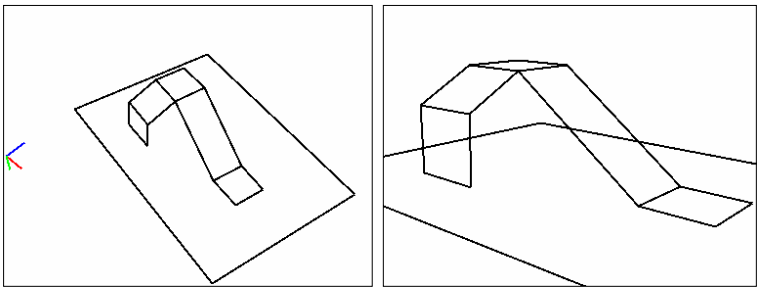


Figure 10: Views of the 3D object used for simulations. The camera frame is represented as follows: $\vec{x}$ axis in red, $\vec{y}$ axis in green, and $\vec{z}$ axis in blue

of the camera during the navigation is presented on Figures 13 and 14. On the second figure, the pose of the robotic system (curves) is compared to the position 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 each image from the path, the curves would pass through the crosses. This clearly not the case.

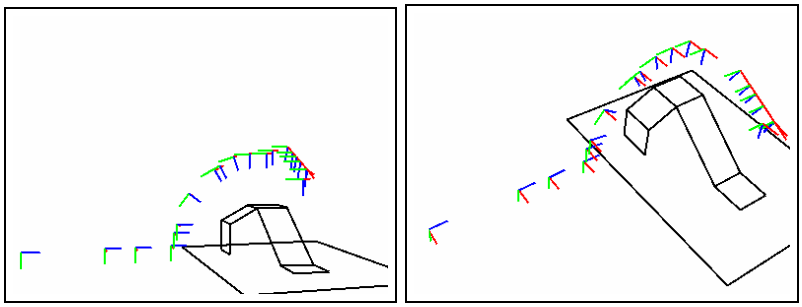


Figure 11: Exp. 1: Positions associated with the image path

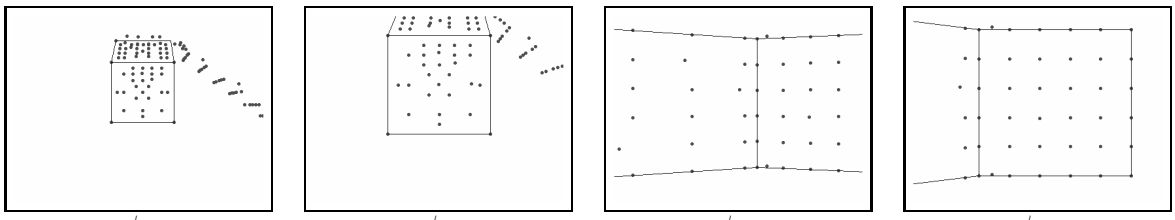


Figure 12: Exp. 1: Examples of images from the path (ψ_0: initial image, ψ_{19}: desired one)

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 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 figure 15, the initial projection of the object is close to the left border of the image. As it can be seen on Figure 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 there used to make the object projection area grows, by getting closer to the

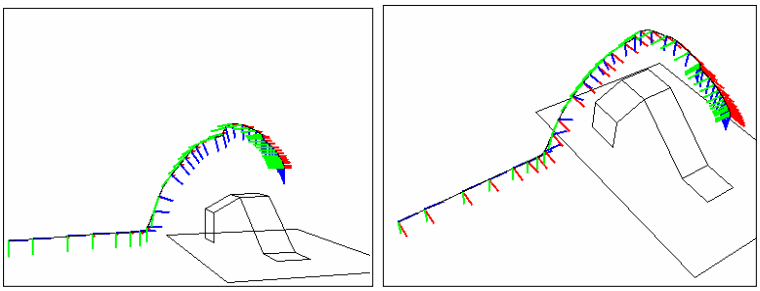


Figure 13: Exp. 1: realized trajectory

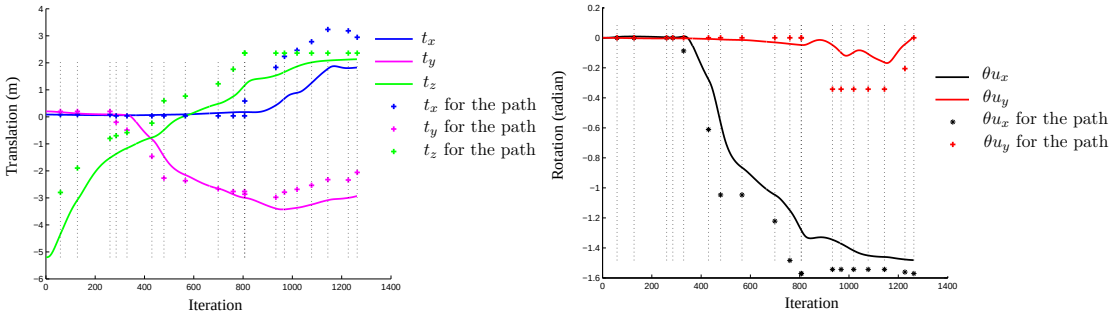


Figure 14: Exp. 1: Position of the camera (translation and orientation) during the navigation. Vertical lines indicate a change of interest set $\mathcal{M}_i$ used to control the system. Crosses indicate the pose associated to the view ψ_{i+1} of the set $\mathcal{M}_i$ just before the change of interest set. The robotic system does not converge towards each of these positions.

object. But in the same time, the object gets closer to the image borders. The first visual measure is thus used to ensure that the object stays into the camera field of view, which is achieved thanks to translation along $\vec{x}$ axis and rotation around $\vec{y}$ axis.

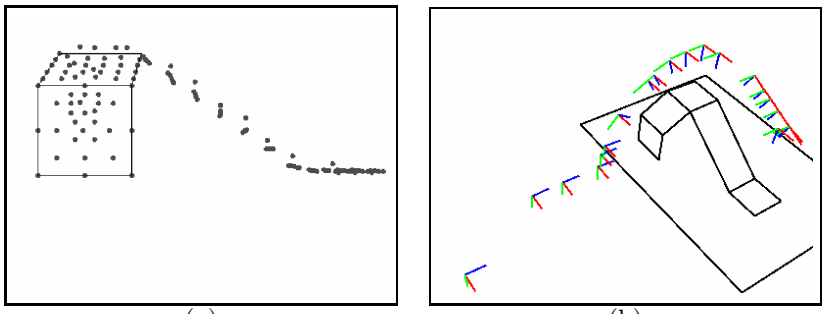
3.2 Set up two: robotic system moving on a plane

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 here controlled with two inputs: one for the translation along $\vec{z}$ axis, and one for the rotation around $\vec{y}$ axis. 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.

Figure 17(a) represents the image path. One can notice on 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 here obvious that the robotic system does not need to reach these positions for performing its navigation task.

Figures 17(b) and 18 describe the trajectory realized by the robotic system. It can be seen that it does not reach the position associated to view 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.



(a) (b)

Figure 15: Exp. 2: (a) inaitial image, (b) image path positions

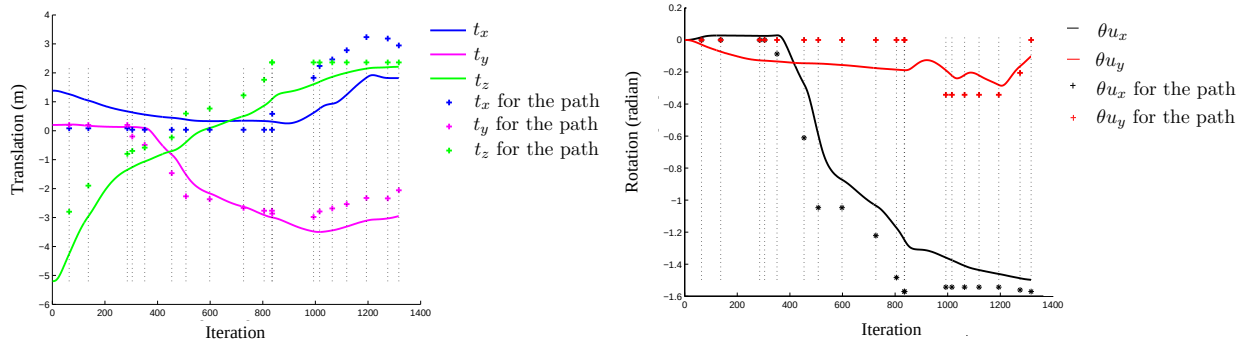


Figure 16: Exp. 2: robotic system positions and orientations during navigation

3.3 Experiments on a robot arm

The next experiments have been realized on a six degrees of freedom robot arm 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 can not go in a straight way from the initial position to the desired one. Images extracted from the database and defining the path to perform are shown in Fig. 19.

When considering a planar scene, the image transfer presented in section 2.2 is much more simple. 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 knowledge of this matrix is sufficient to perform the point transfer. The combination of homographies is also straightforward. Thus, if we consider that the points correspondences between the images ψ_t and ψ_i enable 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 field of view. It has also been used for creating the Figure 20, in which all 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.

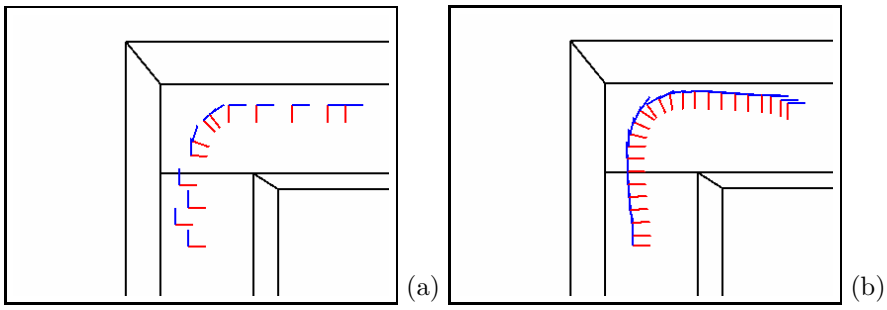


Figure 17: Exp. 3: image path positions and realized trajectory

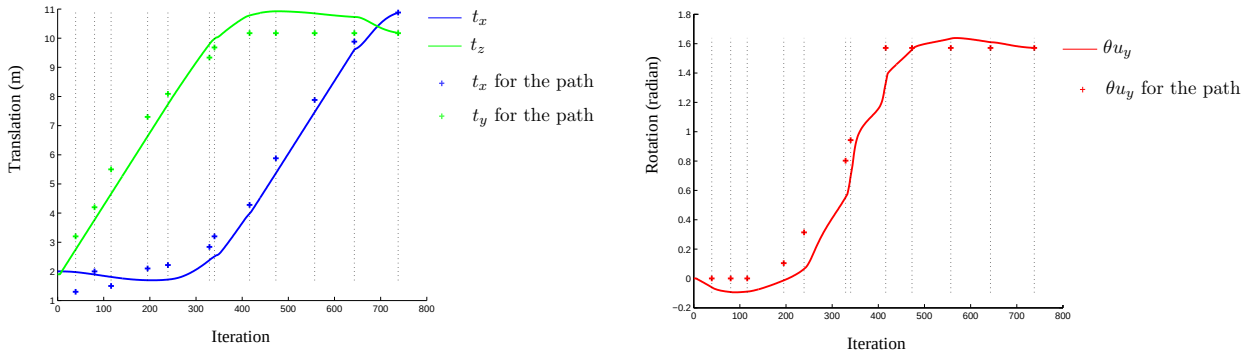


Figure 18: Exp. 3: robotic system positions and orientations during navigation

Figure 21 presents the trajectory of the principal point of the camera during the navigation. Enforcing points of the next scheduled set $\mathcal{M}_i$ to enter inside the camera field of view is sufficient to perform the navigation without reaching the pose of each image of the path. Figure 22 compares the obtained $2D$ trajectory with two other approaches. The first method is a succession 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 of the path as its new desired position. In the second method [Mezouar et al., 2002], 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 of the path is then considered as the desired one. Therefore, the robot no longer converges towards each image of the sequence (as we can see in Fig. 22), but it is still dependent to the intermediary poses.

As one can see, while ensuring that the robotic system stays in areas where enough points are visible, the method proposed manages to realize a shorter trajectory. Furthermore, this trajectory is less dependent to the poses of the images from the path. Indeed, in the second objective function $\mathcal{V}_{\mathbf{s}}$ (see 2.3.3), all the points considered are projected in 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 position measured in the next image of the path. The motion performed is thus naturally dependent to the pose between this view and the current one.

In the next experiment, a 180 degrees rotation is applied to images ψ_1 and ψ_5 of the path. Performing this path with the first approach constraints 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 total convergence towards the

Figure 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.

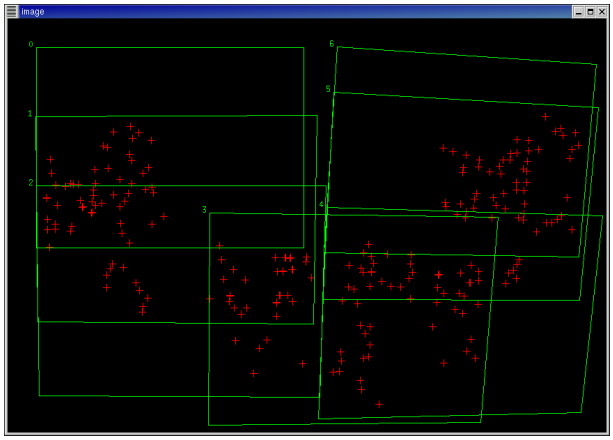


Figure 20: Exp. 4: points and image borders projected onto the first image plane

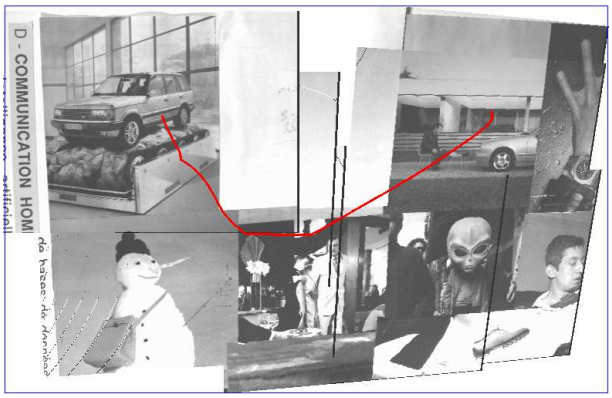


Figure 21: Exp. 4: Principal point trajectory projected onto the first image plane

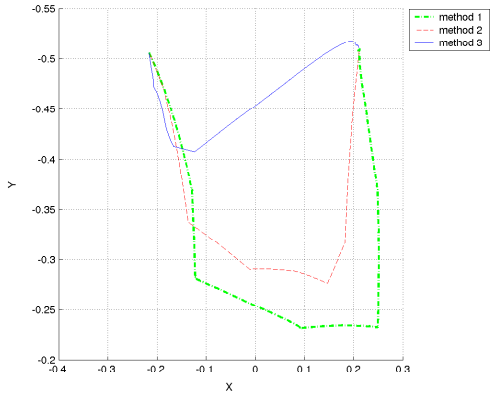


Figure 22: Exp. 4: comparison of $2d$ robot trajectories for the path defined by Fig. 19. Method 1: iterative classical visual servoing with convergence towards each intermediary images. Method 2: the current visual servoing is skipped when enough points from the next view are visible. Method 3: method proposed in this paper. Our approach gives the shortest path.

intermediary images, realizes anyway a part of these rotations. 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 at all our approach.

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, 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 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 a confidence interval. Experiments realized in simulations and with a real robotic system have

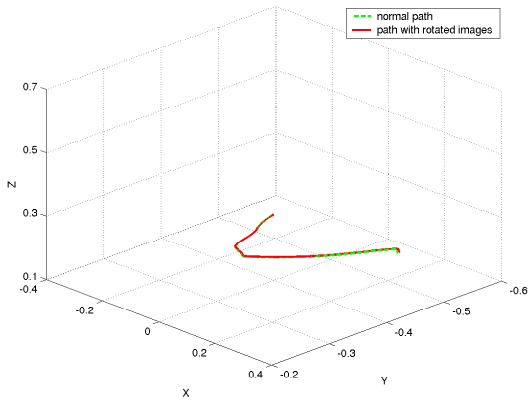


Figure 23: Exp. 4: Robot trajectories compared (path defined by Fig. 19 and the same with rotated images). The two trajectories are equivalent, since the robotic system uses the feature positions in the current frame to control its motion, and not the one observed within the images from the path. The rotated images do not disturb the control law.

shown the validity of the proposed approach.

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

References

[Argyros et al., 2001] Argyros, A., Bekris, C., and Orphanoudakis, S. (2001). Robot homing based on corner tracking in a sequence of panoramic views. In IEEE *Conf. on Computer Vision and Pattern Recognition*, pages 3–10, Kauai, USA.

[Blanc et al., 2005] Blanc, G., Mezouar, Y., and Martinet, P. (2005). Indoor navigation of a wheeled mobile robot along visual routes. In IEEE *Inter. Conf. on Robotics and Automation*, Barcelona, Spain.

[Burschka and Hager, 2001] Burschka, D. and Hager, G. D. (2001). Vision-based control of mobile robots. In IEEE *Inter. Conf. on Robotics and Automation*, pages 1707–1713, Seoul, South Korea.

[Cobzas et al., 2003] Cobzas, D., Zhang, H., and Jagersand, M. (2003). Image-based localization with depth-enhanced image map. In IEEE *Inter. Conf. on Robotics and Automation*, pages 1570–1575, Taipeh, Taiwan.

[Dao et al., 2003] Dao, N. X., You, B. J., Oh, S. R., and Hwangbo, M. (2003). Visual self-localization for indoor mobile robots using natural lines. In IEEE *Inter. Conf. on Intelligent Robots and Systems*, pages 1252–1255, Las Vegas, USA.

[Davison, 2003] Davison, A. (2003). Real-time simultaneous localisation and mapping with a single camera. In IEEE *Inter. Conf. on Computer Vision*, Nice, France.

[De La Torre and Black, 2001] De La Torre, F. and Black, M. J. (2001). Robust principal component analysis for computer vision. In IEEE *Inter. Conf. on Computer Vision*, volume 1, pages 362–369, Vancouver, Canada.

[Espiau et al., 1992] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. IEEE *Trans. on Robotics and Automation*, 8(3):313–326.

[Faugeras and Lustman, 1988] Faugeras, O. and Lustman, F. (1988). Motion and structure from motion in a piecewise planar environment. *Inter. Journal of Pattern Recognition and Artificial Intelligence*, 2:485–508.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey Vision Conf.*, pages 147–151.

[Hartley and Zisserman, 2000] Hartley, R. and Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge University Press, England.

[Jin et al., 2001] Jin, H., Favaro, P., and Soatto, S. (2001). Real-time feature tracking and outlier rejection with changes in illumination. In IEEE *Inter. Conf. on Computer Vision*, volume 1, pages 684–689, Vancouver, Canada.

[Jones et al., 1997] Jones, S., Andersen, C., and Crowley, J. L. (1997). Appearance based process for visual navigation. In IEEE *Inter. Conf. on Intelligent Robots and Systems*, volume 2, pages 551–557, Grenoble, France.

[Košecká et al., 2003] Košecká, J., Zhou, L., Barber, P., and Duric, Z. (2003). Qualitative image based localization in indoor environments. In IEEE *Conf. on Computer Vision and Pattern Recognition*, pages 3–10, Madison, USA.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Inter. Journal of Computer Vision*, 60(2):91–110.

[Malis and Chaumette, 2000] Malis, E. and Chaumette, F. (2000). 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *Inter. Journal of Computer Vision*, 37(1):79–97.

[Matsumoto et al., 2000] Matsumoto, Y., Inaba, M., and Inoue, H. (2000). View-based approach to robot navigation. In IEEE *Inter. Conf. on Intelligent Robots and Systems*, pages 1702–1708, Takamatsu, Japan.

[Mezouar et al., 2002] Mezouar, Y., Remazeilles, A., Gros, P., and Chaumette, F. (2002). Image interpolation for image-based control under large displacement. In IEEE *Inter. Conf. on Robotics and Automation*, volume 3, pages 3787–3794, Washington, USA.

[Rasmussen and Hager, 1996] Rasmussen, C. and Hager, G. (1996). Robot navigation using image sequences. In *Nat. Conf. on Artificial Intelligence*, volume 2, pages 938–943, Portland, USA.

[Remazeilles et al., 2005] Remazeilles, A., Chaumette, F., and Gros, F. (2005). Image based robot navigation in 3d environments. In *Int. Symp. on Optomechatronic Technologies, ISOT'05*.

[Remazeilles et al., 2004] Remazeilles, A., Chaumette, F., and Gros, P. (2004). Robot motion control from a visual memory. In IEEE *Inter. Conf. on Robotics and Automation*, volume 4, pages 4695–4700, New Orleans, USA.

[Royer et al., 2004] Royer, E., Lhuiller, M., Dhome, M., and Chateau, T. (2004). Towards an alternative gps sensor in dense urban environment from visual memory. In *British Machine Vision Conference*, London, England.

[Se et al., 2002] Se, S., Lowe, D., and Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Inter. Journal of Robotics Research*, 21(8):735–758.

[Shashua and Navab, 1996] Shashua, A. and Navab, N. (1996). Relative affine structure: Canonical model for $3d$ from $2d$ geometry and applications. IEEE *Trans. on Pattern Analysis and Machine Intelligence*, 18(9):873–883.

[Tahri and Chaumette, 2005] Tahri, O. and Chaumette, F. (2005). Point-based and region-based image moments for visual servoing of planar objects. IEEE *Trans. on Robotics*, 21(6):1116–1127.

[Thrun et al., 2000] Thrun, S., Burgard, W., and Fox, D. (2000). A real time algorithm for mobile robot mapping with applications to multi-robot and $3d$ mapping. In IEEE *Inter. Conf. on Robotics and Automation*, pages 321–328, San Francisco, USA.

[Zhang et al., 1994] Zhang, Z., Deriche, R., Luong, Q., and Faugeras, O. (1994). A robust approach to image matching: Recovery of the epipolar geometry. *Inter. Symposium of Young Investigators on Information-Computer-Control*.

[Zhou et al., 2003] Zhou, C., Wei, Y., and Tan, T. (2003). Mobile robot self-localization based on global visual appearance features. In IEEE *Inter. Conf. on Robotics and Automation*, pages 1271–1276, Taipeh, Taiwan.