HAL
archives-ouvertes.fr

Robot motion control from a visual memory

A. Remazeilles, François Chaumette, Patrick Gros

▶ **To cite this version:**

A. Remazeilles, François Chaumette, Patrick Gros. Robot motion control from a visual memory. IEEE Int. Conf. on Robotics and Automation, ICRA'04, 2004, New Orleans, Louisiana, United States. pp.4695-4700. inria-00352036

HAL Id: inria-00352036
https://hal.inria.fr/inria-00352036

Submitted on 12 Jan 2009

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

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

Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Robot motion control from a visual memory

Anthony Remazeilles, François Chaumette, Patrick Gros
IRISA - INRIA Rennes
Campus de Beaulieu, 35042 Rennes Cedex, France
Email: Firstname.Name@irisa.fr

Abstract— **This article presents a new approach for robot motion control, using images acquired by an on-board camera. A particularity of this method is that it can avoid reconstructing the entire scene without limiting the displacements possible. To achieve this, an image base of the environment is used to describe the navigation space. We extract from this base a sequence of overlapping images which define the zone that the robot must traverse, in order to reach the desired position. Motions are computed on-line using only points of interest extracted from these images. A method based on potential field theory has been adapted in order to ensure a sufficient visibility of these features during the entire motion of the robot. Experimental results obtained on a six degrees of freedom robotic system are presented and confirm the validity of our approach.**

I. INTRODUCTION

This article deals with automatic robot motion determination using visual information provided by an on-board camera. This problem is not new, but is still largely unsolved especially for real environments. Firstly approaches dealing with this problem are based on the classical *"Perception-Decision-Action"* cycle. Usually, those methods need a preliminary reconstruction step of the navigation environment. On the other hand, several works consider a local approach of visual servoing [5], [12]. It consists of minimizing an error measured between a current and a desired visual information. Therefore it is assumed feature matching can be done, which limits possible displacements.

The main objective of this work is to consider very large motions. A typical example (but still futurist) is a vehicle with an on board camera, able to autonomously displace itself to different places of a city. In order to avoid an expensive (in term of complexity and computational time) reconstruction of the environment, we use an image base of the scene, which will help us to define the motion that the robot can do.

Methods using *Perception-Decision-Action* loop can be classified with respect to the representation they make of the robotic navigation space. Some construct a roadmap representing a collection of mono-dimensional curves associated to configurations the robot can reach. Depending on the method used to obtain it, the roadmap is called a visibility graph [8], Voronoi diagram [25], generalized cones [2] or probabilistic map [19]. Other methods divide the navigation space into cells, corresponding to allowed or forbidden regions of the robot configuration space [7], [27]. But those methods rely on a planification step, which means that the entire robot trajectory is generally computed off-line and therefore it is difficult to deal with unpredictable events occurring during the displacement. Potential field methods, originally developed as an online collision avoidance approach [9], [14], work directly in the work space. The robot is treated as a particle under the influence of an artificial potential field defined as the sum of an attractive potential pulling it toward the desired position, and a repulsive one pushing the robot away from undesirable configurations. However it is still supposed that the place topology is perfectly known, which implies that a reconstruction step must be done before.

Furthermore, in the planification methods presented previously, methods used for robot localization are not presented. Methods called *SLAM* (for *Simultaneous Localization and Map building*) propose to localize the robot and to improve the scene model, thanks to robot motions and laser-like sensors [23]. But these methods are not yet able to use a single camera as a sensor. In [24], localization is obtained with an image retrieval system, based on histograms defined in the neighborhood of each image features. Although a camera is used, environments considered are structured, and no autonomous navigation task has been foreseen.

Some works deal in a same formalism with both localization and navigation. In [4], localization is performed by comparing the environment model, obtained during a learning step, and a local one obtained with an on-board sensor. The scene is divided into free convex regions, which enables to consider a navigation task as a sequence of straight line motions. In [13], [16], a camera sensor is used both for localization and navigation. But theses schemes can not manage navigation tasks between two positions defined by two images totally different.

The method proposed in this paper enables the robotic system localization with respect to an available image base of the environment (and not with respect to its $3D$ environment), and then navigation toward a desired position while satisfying an adapted visual constraint. Therefor, an image retrieval is first performed in order to extract from the data base an image sequence. Those images delimit the area of the whole environment the robot is allowed to traverse to reach its goal. A control law based on potential functions is then executed to move the robot. In [18], a planification method with a sequence of images has been introduced. It will be shown why this new scheme is more efficient.

The next section presents how the sequence of images is extracted from the database. The adaptation of a potential field method is then presented in Section 3. Finally, Section 4 gives some experimental results that confirm the validity of this approach for planar environments.

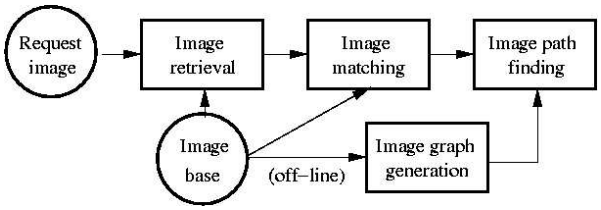

Fig. 1. Successive steps for extracting an image sequence

II. IMAGE SEQUENCE SELECTION

This section describes how a navigation task can be defined in term of $2D$ images. The initial position is defined by the image acquired by the camera before the motion, and the desired position by the image the camera has to obtain at the end of the motion. Those two images will be referred in the rest of this paper as the initial image and the desired one.

The image base corresponds to a set of views describing the whole environment of navigation. When a navigation task is specified, a collection of views, which visually define the scene the camera should observe during the motion, is selected from the base. That means that common features must exist between each couple of consecutive images of the sequence.

Operations presented in this section are carried out before any movement takes place. The first step, construction of the base graph, is done offline, only one time. The two next stages are applied for each motion task, i.e. when a desired image is specified. Fig. 1 summarizes the successive operations.

A. Offline stage: construction of the base graph

We consider here that an image base of the environment has been acquired. In order to define a relationship between those images, a robust matching algorithm [26] is used. It provides, for each couple of images, points of interest that are in correspondence. Those points correspond to high curvatures of the gray scale image [11].

The graph is then generated as follows:

- each node corresponds to one image of the base,
- an edge indicates that at least 4 points have been matched between the two corresponding images. This edge is weighted by the inverse of the number of matched points.

Indeed, the more correspondences images have, the more the motion between the associated positions is likely to be easy, and well controlled. This weighted graph enables to define an *image path* between any couple of images from the base.

B. Image retrieval

Once a navigation task is defined, the first operation consists of linking the initial and desired images with the image base. Therefor, the nearest images to the initial image, and the nearest to the desired one are searched in the base. A content based retrieval system is used, without doing a robust matching with the entire database (which would be too much time consuming). It is reminded that two retrievals are successively done: one for the initial image, and one for the desired one.

More precisely, each point of interest from images of the base are characterized by a descriptor. We used photometric invariants, vectors that remain the same under translation, rotation, scaling changes and illumination variations [20]. The same descriptors are computed for the requested image.

The retrieval consists then of a k-nearest-neighbor-search: each request descriptor gives a vote to the k nearest descriptors from the base (euclidean distance is used). The images having the most votes are the nearest images. The interested reader could refer to [1], [20] for more details on image retrieval.

C. Determination of the image path

Initial and desired images are matched with their most similar images in the base in order to complete the graph. Then, Dijkstra's shortest path algorithm [3] extracts from the base an image path linking the two request images. This methodology assures that:

- consecutive images contain enough common features,
- the selected path is the shortest one, with respect to the weighting system used.

Fig. 4 presents an example of an image sequence extracted from the base. It can be seen that each couple of images has a common region, which ensures that the environment between the initial and desired position is correctly defined.

III. ROBOT MOTION DETERMINATION

Here it is explained how robot motions can be computed with the image sequence. In [18] a robot trajectory is obtained during an offline planification step. It is then followed using an image-based visual servoing which ensures that the robot converges to each image successively (which is constraining and in fact useless).

In our method, the motion is computed on-line, for each image acquired by the camera and without any planification step, which gives to this method a higher reactivity. Indeed, unexpected exterior events occurring during the motion (like a moving obstacle provoking an occlusion) could be easily taken into account within the scheme proposed. Furthermore, each intermediate image need not be reached exactly by the robot during its motion. The image base is employed here to describe the environment in which the robot is likely to move. But this base can not provide any particular motion task with the best intermediary positions.

The $2D$ positions of the points matched between these images are the only information used. Successive matched sets between images of the sequence inform whether or not features should become visible or disappear during the motion. Therefore robot motions will consist of making features initially out of the field of view, become visible, until the robot reaches the desired position. In order to do this, we use a potential field approach, using an original attractive potential.

Reprojection from the sequence of images is used in order to know the position of features not yet visible on the image plane. Therefore the next section reminds some projective geometry notions. The method proposed is then presented.

A. N images geometry and notations

Let us consider two views ψ_1 and ψ_2 of a planar scene. This plane Π is represented in the second frame $\mathcal{F}_2$ by the vector $\pi^T = [\mathbf{n}_2 \ -d_2]$, where $\mathbf{n}_2$ is its normal vector, and d_2 the orthogonal distance between the plane and the optical center. A $3D$ point $P_j \in \Pi$ is projected under perspective projection onto the two images on points measured in pixel $p_{1,j} = [u_1 \ v_1 \ 1]^T$ and $p_{2,j} = [u_2 \ v_2 \ 1]^T$. These projections are linked by the transformation [6]: $p_1 \propto {}^1\mathbf{G}_2 p_2$ where ${}^1\mathbf{G}_2$ is a collineation matrix. It corresponds to a projective frame transformation from view ψ_2 to ψ_1 ($\propto$ is the equality up to a factor). This matrix can be estimated with several methods. Four points are necessary in the general case, or even three if the epipolar geometry is already known [6], [21].

If we suppose the camera internal parameters $\mathbf{K}$ known, the collineation ${}^1\mathbf{H}_2 = \mathbf{K}^{-1}{}^1\mathbf{G}_2\mathbf{K}$ can be decomposed in:

$$ {}^1\mathbf{H}_2 = {}^1\mathbf{R}_2 + \frac{{}^1\mathbf{t}_2}{d_2}\mathbf{n}_2^T, $$

where $({}^1\mathbf{R}_2 \ {}^1\mathbf{t}_2)$ is the rigid motion between the two frames $\mathcal{F}_1$ and $\mathcal{F}_2$. ${}^1\mathbf{R}_2$ and ${}^1\mathbf{t}_2$ (up to a factor) can be extracted from this collineation matrix [6]. It is also possible to determine the ratio ρ_j, defined for each couple of projection between the depth Z_j of the $3D$ point and d_2 [17]:

$$ \rho_j = \frac{Z_j}{d_2} = \frac{1 + \mathbf{n}_2^T {}^1\mathbf{R}_2^T ({}^1\mathbf{t}_2/d_2)}{\mathbf{n}_2^T {}^1\mathbf{R}_2^T \mathbf{K}^{-1} p_1} \tag{1} $$

Let us now consider a set of $N+1$ images ψ_i ($i \in [0, N]$). ψ_0 is the initial image, and ψ_N the desired one. $m_{i,j}$ is the metric coordinates of the projection onto the image plane ψ_i of the $3D$ point P_j. $\mathcal{M}_i$ is the feature set matched between views ψ_i and ψ_{i+1}. A couple of projections from $\mathcal{M}_i$ is noted $(m_{i,j}, m_{i+1,j})$. These matching sets enable to obtain N metric collineation matrices: ${}^0\mathbf{H}_1, \ldots, {}^{N-1}\mathbf{H}_N$. By composing these collineations, a relation between any pair of images is obtained:

$$ m_{i,j} \propto \prod_{l=i}^{k-1} {}^l\mathbf{H}_{l+1} m_{k,j} = {}^i\mathbf{H}_k m_{k,j}, \tag{2} $$

with $i < k \leq N$. This composition enables to predict a feature position in the current image plane even if the considered point is not yet visible (a similar image transfer can be found in [10]). Feature projections that are nearby the camera field of view can then be detected, and the robot can move in order to make them enter into the visibility area.

A projection $m_{t,j} \propto (u, v, 1)$ is said visible if $u \in [u_m \ u_M]$ and $v \in [v_m \ v_M]$, where u_m, u_M, v_m and v_M define a frame in the current image ψ_t. We note $\mathcal{C}_{free}$ this visibility area. The set $\mathcal{I}_i$ corresponds to the $3D$ points that have been detected in the image ψ_i of the path. Points that are visible in the current image ψ_t are noted $s_{t,j}$. V_i is the set of features from the image ψ_i of the path that are visible in ψ_t, which means:

$$ s_{t,j} \in V_i \Longleftrightarrow P_j \in \mathcal{I}_i $$

The robot configuration is represented by a vector $\mathcal{X}$ in the configuration space $\mathcal{W}$. As the $3D$ scene model is not known, the partial parameterization $\mathcal{X}_k = \left[{}^k\mathbf{t}_{d_N N}, \mathbf{u}\theta\right]$ is used. From the collineation ${}^k\mathbf{H}_N$ it is possible to compute ${}^t\mathbf{R}_N$, and ${}^k\mathbf{t}_{d_N N} = {}^k\mathbf{t}_N / d_N$. The normalized rotation axis $\mathbf{u}$ and the angle of rotation θ are deduced from ${}^t\mathbf{R}_N$.

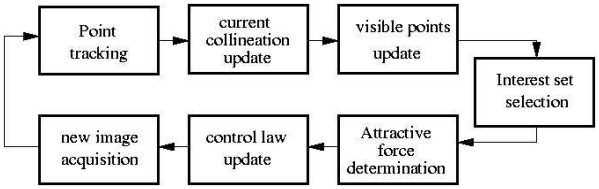

Fig. 2. Loop realized to compute the robot motion

B. Proposed methodology

The method proposed consists in making enter into the current image frame feature projections of the next images of the path that are not yet visible. Step by step, the robot moves thus toward its desired position. This section presents the successive operations done for each image acquired by the camera. Fig. 2 summarizes these steps.

Before the motion starts, the collineations linking the projections between each successive couple of images, ${}^i\mathbf{H}_{i+1}$ ($i \in [0, N-1]$), are computed. All known points are then projected onto the first and initial image plane ψ_0:

$$ m_{0,j} \propto {}^0\mathbf{H}_k m_{k,j}, $$

with $k \in [1, N-1]$ and the matrices ${}^0\mathbf{H}_k$ obtained by collineation composition. Features that belong to $\mathcal{C}_{free}$ will be tracked in the next image. From theses visible points, noted $s_{0,j}$, sets V_i can also be initialized.

1) **Feature tracking:** in ψ_{t-1}, the position of a visible feature set $s_{t-1,j}$ is known. The well-known Shi-Tomasi-Kanade point tracker [22] permits to update their position $s_{t,j}$ in the current image ψ_t. Some points may get out the free area $\mathcal{C}_{free}$. They are kept for the moment to compute the current collineation (those points are out of $\mathcal{C}_{free}$, but still in the current frame). The following step will take care of them, as well as points that are entering into the free area thanks to robot motions.

2) **Current collineation determination:** collineations between the current image and images ψ_i of the forthcoming path are computed. Two possibilities raise:

- $card\,(V_i) \geq N_h$
- $card\,(V_i) < N_h$,

where $card(S)$ is the number of elements of the set S and N_h is the minimal number of points needed to compute a collineation matrix. In the first case, the collineation is directly obtained, by resolving the following system:

$$ s_{t,j} \propto {}^t\mathbf{H}_i m_{i,j} \ , \forall s_{t,j} \in V_i \tag{3} $$

In the second case, the image content is not sufficient to directly compute the collineation. But if we consider that the collineation ${}^t\mathbf{H}_l$ between ψ_t and the frame ψ_l of the path has

been obtained with (3), the collineation between the current frame and ψ_i can be deduced from:

$$^t\mathbf{H}_i = {}^t\mathbf{H}_l {}^l\mathbf{H}_i, \tag{4}$$

where $l < i < N$ and $^l\mathbf{H}_i$ obtained by composition of collineations defined on the image path.

3) **Visible features update:** with these collineations, features $m_{i,j}$ can be projected onto the current image plane from the image ψ_i they belong to. A set $m_{t,j}$ is obtained:

$$m_{t,j} \propto {}^t\mathbf{H}_i m_{i,j} \tag{5}$$

This reprojection concerns only points that are not yet visible. Indeed, as long as collineations are computed from matched points and/or by collineation composition, no more precision in term of position for the tracking can be obtained by reprojecting already known points.

Points $s_{t,j}$ tracked between previous and current views that no longer belong to the free area are removed. Points obtained with (5) verifying $m_{t,j} \in \mathcal{C}_{free}$ are added to $s_{t,j}$. Sets V_i are also updated, with respect to the novel set $s_{t,j}$.

4) **Interest point set selection:** we are looking among all the sets of matching $\mathcal{M}_i$ defined onto the path the one $\mathcal{M}_{i^*}$ that will be used to define the robot motion. The selection criterion is that its point projections must be close to the free area, and at the same time, this set must be the furthest with respect to the path. Therefore, we select among all the sets $\mathcal{M}_i^*$ verifying:

$$card\left(m_{i^*,j} | m_{i^*,j} \in V_{i^*} \wedge (m_{i^*,j}, m_{i^*+1,j}) \in \mathcal{M}_{i^*}\right) \geq N_M$$

the one with the rank i^* maximal (N_M is a threshold).

5) **Attractive force computation:** the attractive force is defined by [17]:

$$\mathbf{F_f}(\mathbf{x}) = -\epsilon \left(\frac{\partial \mathbf{f}}{\partial \mathcal{X}}\right)^+ \vec{\nabla}_\mathbf{f}^T \mathcal{V}_\mathbf{f},$$

where $\mathbf{f}$ is a derivable function onto the whole configuration space $\mathcal{W}$ and $\vec{\nabla}_\mathbf{f}^T \mathcal{V}_\mathbf{f}$ is the gradient of a potential function $\mathcal{V}_\mathbf{f} = \mathcal{V}(\mathbf{f}(\mathcal{X}))$. ϵ is a positive gain used to fix the amplitude of the force.

The attractive potential $\mathcal{V}_\mathbf{f}$ is defined onto the image plane, in order to attract into the field of view features from $\mathcal{M}_{i^*}$ that are not yet visible. We propose the following potential:

$$\mathcal{V}_s = \sum_j \mathcal{V}_s(s_j), \tag{6}$$

with:

$$\begin{aligned}\mathcal{V}_s(s_j) = g(u_j - u_M) + g(v_j - v_M) \\ + g(u_m - u_j) + g(v_m - v_j),\end{aligned}$$

and

$$g(x) = x * \left(\pi^{-1} \arctan(k\pi x) + t\right).$$

s_j is the coordinate vector $(u_j\ v_j\ 1)^T$ of the feature concerned, k and t are constants. If all the points s_j project into $\mathcal{C}_{free}$, this potential is null. Fig. 3 shows this potential function in the case of a single point.

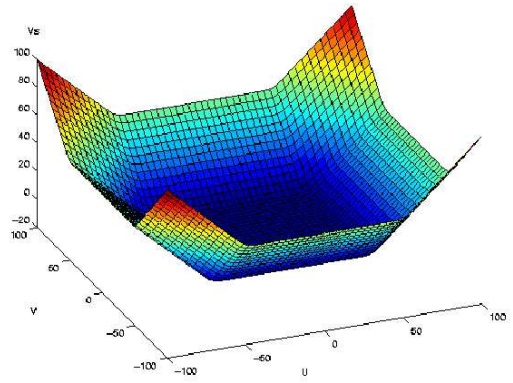

Fig. 3. Visibility constraint based potential function, for a single point

The corresponding attractive force is then:

$$\mathbf{F_s} = -\epsilon \left(\frac{\partial \mathbf{s}}{\partial \mathcal{X}}\right)^+ \vec{\nabla}_\mathbf{s}^T \mathcal{V}_\mathbf{s} = -\epsilon \mathbf{L}^+ \vec{\nabla}_\mathbf{s}^T \mathcal{V}_\mathbf{s},$$

where $\vec{\nabla}_\mathbf{s}^T \mathcal{V}_\mathbf{s}$ is easily obtained from (6) and $\mathbf{L}$ is the interaction matrix related to s [5]. It defines the motion of the image features with respect to the camera velocity $\mathcal{T}_\mathbf{c} : \dot{s} = \mathbf{L}\mathcal{T}_\mathbf{c}$. For a k feature set, the interaction matrix is:

$$\mathbf{L}(\mathbf{s}, \mathbf{Z}) = \left[\ \mathbf{L}^T(\mathbf{p}_1, Z_1) \quad \ldots \quad \mathbf{L}^T(\mathbf{p}_k, Z_k)\ \right]^T, \tag{7}$$

where $\mathbf{L}(\mathbf{p}_i, Z_i)$ is the classical interaction matrix of a point p_i whose depth is Z_i. Considering relations (7) and (1), we obtain [17]:

$$\mathbf{L}(\mathbf{p}, d_{i^*+1}) = \frac{1}{d_{i^*+1}} \begin{bmatrix}\mathbf{S} & \mathbf{Q}\end{bmatrix}$$

where $\mathbf{Q} = \begin{bmatrix}\mathbf{Q}_1^T \ldots \mathbf{Q}_k^T\end{bmatrix}$ and $\mathbf{S} = \begin{bmatrix}\mathbf{S}_1^T \ldots \mathbf{S}_k^T\end{bmatrix}$ are two $2n \times 3$ matrices independent of d_{i*+1}. $\mathbf{S}_j$ and $\mathbf{Q}_j$ are defined as :

$$\mathbf{S}_j = \begin{bmatrix} -\frac{1}{\rho_{t,j}} & 0 & \frac{x}{\rho_{t,j}} \\ 0 & -\frac{1}{\rho_{t,j}} & \frac{y}{\rho_{t,j}} \end{bmatrix} \mathbf{Q}_j = \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix}$$

Ratios $\rho_{t,j}$ are deduced from $^t\mathbf{H}_{i^*+1}$ with (see (1)):

$$\rho_{t,j} = \frac{1 + \mathbf{n}_{i^*+1}^T {}^t\mathbf{R}_{i^*+1}^T ({}^t\mathbf{t}_{i^*+1}/d_{i^*+1})}{\mathbf{n}_{i^*+1}^T {}^t\mathbf{R}_{i^*+1}^T \mathbf{K}^{-1} p_j},$$

where p_j is the projection onto the current image of the point P_j considered ($p_j = s_{t,j}$ if the point is visible, $p_j = p_{t,j}$ otherwise).

6) **Control Law:** the robot velocity is directly servoed in order to move in the direction defined by the previous force:

$$\mathcal{T}_c = \mathbf{F}_s$$

A novel image ψ_{t+1} is then acquired. This scheme is done in a loop-way until enough image features from the desired image ψ_N are in the camera field of view. At this moment, the robot is considered to be close to the desired position.

Fig. 4. Example of an image sequence. a is the initial image, and g is the desired one. Other ones have been automatically extracted from the base.

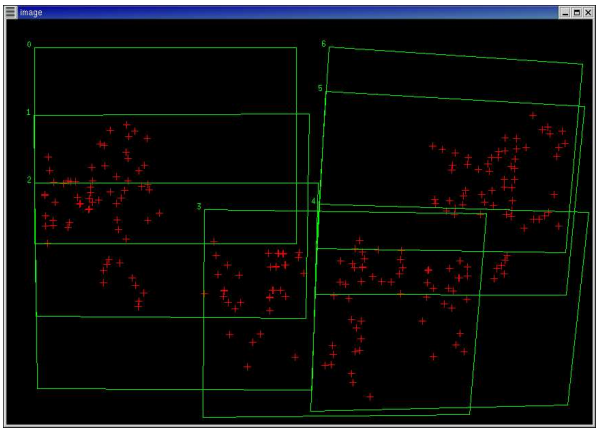

Fig. 5. Points and image borders projected onto the first image plane

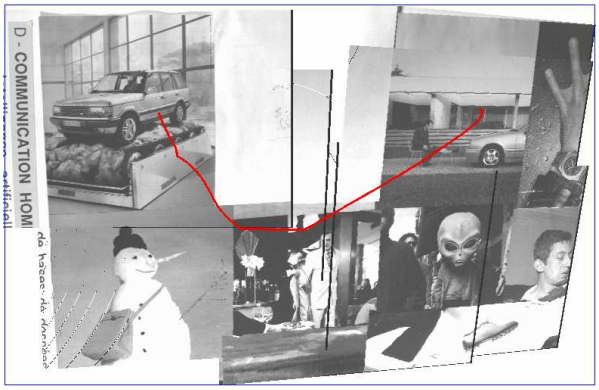

Fig. 6. Principal point trajectory projected onto the first image plane

7) **Desired position convergence:** Once the robot is close to the desired position, a classical attractive force is used until the total system convergence:

$$\mathbf{F}_a = -\epsilon \vec{\nabla}_{\mathbf{a}}^T \mathcal{V}_{\mathbf{a}},$$

where the potential force is a parabolic function:

$$\mathcal{V}_{\mathbf{a}} = \frac{1}{2}\|\mathcal{X}_{\mathbf{t}} - \mathcal{X}^*\|^2.$$

Robot position $\mathcal{X}_{\mathbf{t}}$ is deduced from matrix ${}^t\mathbf{H}_N$, which also enables to detect when points belonging to the last matching set are entering the $\mathcal{C}_{free}$ area. Since $\mathcal{X}^* = \mathbf{0}_{6\times 1}$, the attractive force is nothing but:

$$\mathbf{F}_a = -\epsilon \mathcal{X}_{\mathbf{t}}$$

Robot motion are deduced from $\mathcal{T}_c = \mathbf{F}_a$. This control law exactly corresponds to an hybrid visual servoing [15]. It could be also possible to use an image-based visual servoing, based on the error between the current and desired feature positions.

IV. Experimental results

Experiments presented here were obtained on a six degrees of freedom robot arm, with an on-board camera. The navigation space is a plane on which several photographies are sticked. 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 base and defining the path to perform are shown in Fig. 4.

Fig. 5 presents the reprojection onto the first image plane of the whole interest points of the images. Image borders are also drawn. A large amount of points are not visible in the first view.

A. Obtained trajectory comparison

Fig. 6 shows the trajectory done by the principal point of the camera during the motion. The robot does not reach intermediary positions corresponding to images of the sequence. The $2D$ trajectory is compared to two other methods in Fig. 7. The first method is a planification based on the temporal decomposition of the collineation matrices between each couple of images. The result trajectory is then followed by an image-based visual servoing [18]. The different images of the sequence are in this case intermediary desired positions that the robotic system successively reaches. In the second method, the robot still converges to intermediary positions with an image-based visual servoing, but the current servoing is stopped as soon as enough points defining the next servoing are visible. The next image of the path is then considered as the desired one. Therefore, the robot no longer converges to each image of the sequence (as we can see in Fig. 7), but it is still dependent to the intermediary positions.

The method proposed in this article gives a shorter trajectory, while abiding by the visibility constraint. Moving for making features enter into the camera field of view does not penalize at all the robot motion.

B. Intermediary view positions independence

In order to show the independence of our method to the positions associated to the intermediary images, a 180 degrees rotation were applied to images b and f. Resolution of this path with [18] constraints the robot to make those useless rotations during motions ab, bc, ef and fg. Second method, even if it avoids the total convergence to the intermediary

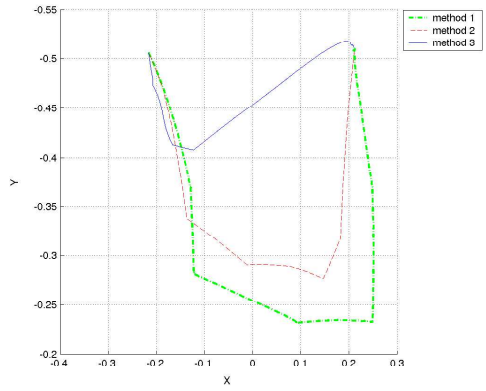

Fig. 7. $2d$ robot trajectory for the path defined by Fig. 4

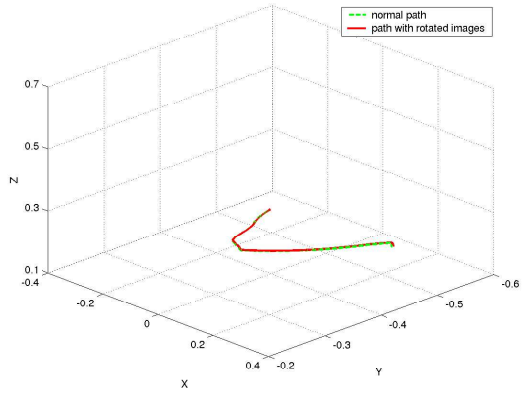

Fig. 8. Robot trajectories compared (path defined by Fig. 4 and the same with rotated images)

images, realizes anyway a part of those rotations. Fig. 8 compares the trajectory for the path without rotation, and the trajectory obtained when views b and f are rotated. The two trajectories are nearly the same, which proves that our method is independent to the positions associated to the image path.

V. Conclusion

This article has presented a novel method for robot motion control, by considering visual information. The path to realize is first described by a sequence of images extracted from a base of the environment. This method, which uses potential field theory, avoids from local minima by only using one attractive force, dealing with the camera attraction toward next image features defined onto the path. Experiments prove also that the trajectory does not depend at all on the positions associated to the images of the sequence. Nevertheless, we can for the moment only consider planar environments. We are therefore working on this point, and looking forward to integrate non holonomic constraints in order to work with a mobile robot.

References

[1] S.A. Berrani, L. Amsaleg, and P. Gros. Approximate searches: k-neighbors+precision. In *ACM International Conference on Information and Knowledge Management*, Louisiane, USA, 2003.

[2] R.A. Brooks. Solving the find-path problem by representing free space as generalized cones. *IEEE Trans. on Systems, Man and Cybernetics*, 13(3):190–197, 1983.

[3] T.H. Cormen, C Stein, R.L. Rivest, and C.E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

[4] J. L. Crowley. Navigation for intelligent mobile robot. IEEE *Journal of Robotics and Automation*, 1(1), 1985.

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

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

[7] B. Faverjon. Obstacle avoidance using an octree in the configuration space of a manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 504–512, Atlanta, Ga., 1984.

[8] S. K. Ghosh and D. M. Mount. An output sensitive algorithm for computing visibility graphs. In *IEEE Symp. on Foundations of Computer Science, Los Angeles*, 1987.

[9] H. Haddad, M. Khatib, S. Lacroix, and R. Chatila. Reactive navigation in outdoor environments using potential fields. In *IEEE Int. Conf. on Robotics and Automation*, pages 1232–1237, Louvain, May 1998.

[10] D. Hager, D. Kriegman, E. Yeh, and C. Rasmussen. Image-based prediction of landmark features for mobile robot navigation. In *IEEE Int. Conf. on Robotics and Automation*, pages 1040–1046, 1997.

[11] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conf.*, pages 147–151, University of Manchester, England, Sept. 1988.

[12] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5):651–670, Oct. 1996.

[13] S. Jones, C. Andersen, and J. L. Crowley. Appearance based processes for visual navigation. *IEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sept. 1997.

[14] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–98, 1986.

[15] E. Malis and F. Chaumette. Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Trans. on Robotics and Automation*, 18(2):176–186, April 2002.

[16] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *IEEE Int. Conf. on Robotics and Automation*, pages 83–88, Minneapolis, 1996.

[17] Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Trans. on Robotics and Automation*, 18(4):534–549, August 2002.

[18] Y. Mezouar, A. Remazeilles, P. Gros, and F. Chaumette. Image interpolation for image-based control under large displacement. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 3787–3794, Washington DC, May 2002.

[19] C. Nissoux, T. Simon, and J-P. Laumond. Visibility based probabilistic roadmaps. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1316–1321, Kyongju, Core, Oct. 1999.

[20] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence*, 19(5):530–534, May 1997.

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

[22] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Vision and Pattern Recognition*, pages 593–600, Seattle, June 1994.

[23] A.C. Victorino, P. Rives, and Borrelly. Localization and map building using a sensor-based control strategy. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 937–942, Takamatsu, Japon, Oct. 2000.

[24] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization for mobile robots using an image retrieval system based on invariant features. In *IEEE Int. Conf. on Robotics and Automation*, Washington DC, May 2002.

[25] C.K. Yap. An $\mathcal{O}(n \log n)$ algorithm for the voronoi diagram of a set of simple curve segments. Technical report, Robotics Laboratory, Courant Institute, New-York University, 1985.

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

[27] D. Zhu and J.C. Latombe. New heuristic algorithms for efficient hierarchical path planning. *IEEE Trans. on Robotics and Automation*, 7:9–20, 1991.