

Contrôle des mouvements d'un robot à l'aide d'une mémoire visuelle

Robot motion control from a visual memory

Anthony Remazeilles

François Chaumette

Patrick Gros

IRISA, Campus de Beaulieu,
35042 Rennes Cedex, France
Prenom.Nom@irisa.fr

Cet article propose une nouvelle approche pour déterminer les mouvements d'un robot en se basant sur les images fournies par une caméra embarquée. Elle a la particularité de s'affranchir d'une étape de reconstruction de l'environnement, sans pour autant limiter les déplacements. Pour cela, nous utilisons une base d'images décrivant l'espace de navigation. De cette base est extraite une série d'images qui définissent la zone que doit traverser le robot pour atteindre la position désirée. Les mouvements du robot sont déterminés en ligne en s'appuyant seulement sur des primitives d'intérêt extraites des images. Une méthode basée sur les champs de potentiel a été adaptée pour assurer une visibilité suffisante de ces primitives tout au long du déplacement. Des résultats expérimentaux obtenus sur un robot cartésien à six degrés de liberté sont présentés et confirment la validité de notre approche.

vision robotique, reconnaissance d'images, champs de potentiel

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 six dof robotic system are presented and confirm the validity of our approach.

robotic vision, image retrieval, potential fields

1 Introduction

Nous nous intéressons dans cet article à la détermination automatique des mouvements d'un robot en utilisant les informations fournies par une caméra embarquée sur ce robot. Cette problématique n'est pas récente mais reste encore ouverte pour des environnements réels de taille significative. L'approche la plus ancienne pour résoudre ce problème est basée sur le cycle *Perception-Décision-Action*. Elle nécessite en général une reconstruction de l'environnement. Par ailleurs, de nombreux travaux portent sur une approche locale d'asservissement visuel [13, 3]. Elle consiste à minimiser l'erreur entre les informations visuelles courantes et désirées. Elle repose donc sur une mise en correspondance, ce qui limite la taille des déplacements qu'il est possible de considérer.

Notre objectif est de réaliser de très grands déplacements. Un exemple typique (et certes encore futuriste) est celui d'un véhicule doté d'une caméra capable de se rendre automatiquement en plusieurs endroits d'une cité. Pour éviter une phase de reconstruction difficile de l'environnement (en terme de complexité et de temps de calcul), nous utilisons une base d'images de l'environnement, qui va nous permettre de définir le chemin à parcourir par le robot.

Les méthodes utilisant le cycle *Perception-Décision-Action* se distinguent notamment par la représentation qu'elles ont de l'espace de navigation du robot. Certaines élaborent une carte de navigation, succession de courbes monodimensionnelles correspondant à des positions de l'espace que peut atteindre le robot. La carte s'appelle graphe de visibilité [9], diagramme de Voronoi [27], cônes généralisées [2] ou encore carte probabiliste [21], suivant la méthode utilisée pour l'obtenir. D'autres techniques décomposent l'espace de navigation sous forme de cellules, zones autorisées ou interdites au robot [7, 29]. Les méthodes basées sur les potentiels [15, 11] travaillent généralement directement dans

l'espace de travail. Le robot est considéré comme une particule sous l'influence d'un champ de potentiel artificiel, défini comme la somme d'un potentiel attractif pour attirer le robot vers la position désirée et d'un potentiel répulsif afin d'éviter les configurations indésirables. Mais toutes ces méthodes supposent que la topologie des lieux est parfaitement connue, ce qui sous-entend qu'une phase de reconstruction est effectuée. De plus, la plupart reposent sur une phase préalable de planification, c'est-à-dire que l'ensemble de la trajectoire est calculé hors-ligne, ce qui rend difficile la prise en compte d'événements imprévus survenant lors du déplacement. Nous avons préféré développer une approche plus réactive, qui ne nécessite pas de reconstruction de l'environnement.

Dans les méthodes de planification présentées ci-dessus, le modèle de la scène est considéré comme connu, et les méthodes de localisation du robot ne sont pas précises. Les techniques appelées SLAM (Simultaneous Localization and Map building) permettent de localiser le robot tout en enrichissant le modèle de la scène grâce au mouvement et à des capteurs de type laser [25]. Mais ces techniques limitent actuellement des espaces intérieurs, très structurés, et ne peuvent encore se satisfaire d'une caméra comme capteur. Dans [26], la localisation d'un robot mobile est assurée par une recherche d'images, basée sur des histogrammes générés sur les images de chaque primitive des images. Bien qu'une caméra soit utilisée, les scènes traitées se limitent aux environnements structurés, et aucun tâche de navigation autonome n'est prévue.

Certains travaux considèrent dans un même formalisme une phase de localisation et de navigation. Dans [4] un modèle de la scène est déduit d'une phase d'apprentissage. La localisation s'effectue en comparant les informations fournies par le capteur un sonar et le modèle appris. Ce modèle est de plus décomposé sous la forme de cellules convexes, ce qui permet de traiter une phase globale de navigation par une succession de sous-tâches (déplacement entre les cellules connectées du chemin). Dans [18, 14], un capteur caméra est utilisé pour la localisation et la navigation d'un véhicule robotisé. Cependant le formalisme utilisé pour la localisation ne permet pas de prendre en compte des déplacements entre deux positions caractérisées par deux images totalement différentes.

La technique que nous proposons permet de situer le système robotique par rapport à une base d'images disponible de l'environnement, et ensuite de réaliser un déplacement vers une position désirée tout en intégrant une contrainte visuelle adaptée. Pour cela, une recherche sur la base est tout d'abord effectuée afin d'en extraire une série d'images. Celles-ci délimitent la zone de l'environnement total que doit traverser le robot pour se diriger vers son but. Une loi de commande basée sur l'approche des fonctions de potentiels [15], est ensuite exécutée pour réaliser le déplacement. Dans [20], une méthode de planification a part d'une série d'images a été proposée. Nous montrerons dans cet article en quoi notre nouvelle méthode est plus performante.

La section suivante présente comment est extraite de la base de l'environnement une série d'images. La section détaille ensuite comment la méthode des champs de potentiels peut être adaptée pour calculer le mouvement du robot. Enfin la section donne des résultats expérimentaux qui valident cette approche dans le cas de scènes planes.

2 Sélection d'une série d'images de la base

Cette partie présente comment l'image acquise avant le début des mouvements du robot et celle désirée à la fin du déplacement sont mises en relation. On cherche dans la base une série d'images décrivant la scène que doit observer la caméra du robot durant le déplacement. Pour cela, chaque couple d'images successives du chemin doit posséder des primitives en commun.

La figure 1 décrit les différentes étapes nécessaires pour cette extraction. Tout d'abord, aussitôt la base de l'environnement acquiert les images qu'elle contient sont traitées afin de générer un graphe évalué en fonction des images en correspondance entre chaque couple d'images de la base. Suite à la sélection d'une image à atteindre, que nous appellerons image désirée, et avec l'image initiale fournie par la caméra avant le début du déplacement, les opérations effectuées sont les suivantes :

- recherche des images de la base les plus proches de l'image initiale et désirée par un algorithme d'indexation par le contenu. A ce niveau, les images initiales et désirées ferment donc des images requêtées à la base,
- appariement des images les plus proches avec deux images requêtées afin de compléter le graphe de l'application,
- détermination de la série d'images par recherche du plus court chemin dans le graphe.

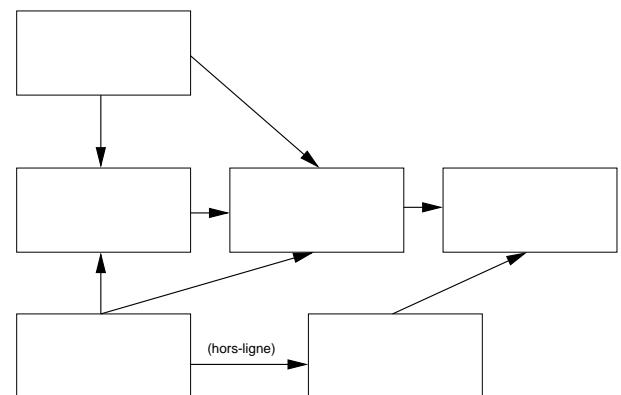


FIG. 1 – Opérations pour la détermination du chemin d'images

La section suivante présente la construction du graphe ainsi qu'un algorithme d'appariement que nous utiliserons. La section détaille la phase d'indexation, et la section montre comment est obtenue une série d'images.

3 Détermination des mouvements du robot

Cette partie présente comment est déterminé le mouvement du robot à partir de la série d'images extraite de la base de données. Contrairement à la méthode présentée dans [20] (une planification hors ligne de trajectoires dans l'image suivie ensuite par asservissement visuel)*D*e mouvement est calculé en ligne, pour chaque image acquise par la caméra sans phase de planification. L'intérêt d'une telle méthode est sa plus grande réactivité. En effet, la prise en compte d'événements extérieurs nécessitant une modification du mouvement du robot (tels l'arrivée d'obstacles générant une occultation) pourrait alors directement être intégrée dans l'algorithme que nous proposons ci-dessous. De même, dans [20], le robot convergeait successivement vers toutes les images extraites de la base, ce qui n'a pas de véritable intérêt. Atteindre chaque image du chemin peut en effet entraîner le robot à effectuer des mouvements inutiles, comme des changements d'orientation, seulement parce que les images du chemin le spécifient et non pas parce qu'ils sont nécessaires pour atteindre la position désirée. La base d'images a pour rôle de décrire le contenu de l'environnement de navigation, et non pas de définir pour un déplacement particulier les positions intermédiaires les plus adaptées. Nous souhaitons donc que les mouvements du robot ne soient pas dépendants des positions associées aux images du chemin.

L'information que nous considérons est seulement la position des primitives mises en correspondance entre ces images. En effet, les ensembles d'appariements réalisés sur les images du chemin nous informent sur l'évolution des primitives visibles par la caméra. Par exemple, les points mis en correspondance sur le début du chemin sont amenés à sortir du champ de vue de la caméra au cours du déplacement, alors que les primitives de la fin du chemin vont elles devenir visibles lorsque le robot se rapprochera de sa position désirée. Le robot va donc se déplacer pour que les primitives extraites du chemin qui ne sont pas encore dans le champ de vision de la caméra soient visibles, et ce jusqu'à ce que le robot atteigne sa position désirée. Pour cela, nous utilisons l'approche des fonctions de potentiel basée sur la définition d'une nouvelle force attractive.

Pour déterminer la position sur le plan image des primitives qui ne sont pas encore visibles, nous allons effectuer des reprojctions depuis les images du chemin. La section suivante effectue donc un rappel de géométrie projective (projective car nous ne faisons pas de reconstruction 3D). Nous présenterons ensuite la méthode que nous proposons.

N

Nous considérons tout d'abord deux prises de vue et d'une même scène. Un plan de référence Π est représenté dans le repère de la deuxième prise de vue \mathcal{F} par le vecteur $\pi^T = [\mathbf{n} \ -d]$, où \mathbf{n} est son vecteur unitaire et d la distance orthogonale du plan au centre optique de la deuxième caméra. Si l'on considère une scène plane, alors les coor-

données pixelliques des projections sur les deux images, $p = [u \ v \ 1]$ et $p = [u \ v \ 1]$, d'un point de la scène $P \in \Pi$ sont liées par la transformation projective [6] :

$$p \propto \mathbf{G}_\pi p,$$

où \mathbf{G}_π est la collinéation en pixel, relative au plan Π . Elle correspond à un changement de repère projectif de l'image vers (\propto représente ici l'égalité à un facteur d'échelle près).

Différentes techniques permettent de calculer cette matrice. Huit points sont nécessaires dans le cas général, quatre seulement si les points sont coplanaires, ou encore trois points non collinaires si la géométrie épipolaire a déjà été déterminée [6, 17, 23].

Connaissant les paramètres intrinsèques \mathbf{K} de la caméra, il est possible de remonter à la collinéation exprimée en mètre :

$$\mathbf{H}_\pi = \mathbf{K}^{-1} \mathbf{G}_\pi \mathbf{K}$$

Cette collinéation se décompose sous la forme :

$$\mathbf{H}_\pi = \mathbf{R} + \frac{\mathbf{t}}{d} \mathbf{n}^T,$$

(\mathbf{R} \mathbf{t}) représentant le changement de repère entre les deux prises de vue \mathcal{F} et \mathcal{F}' . La rotation et la translation à un facteur d'échelle près peuvent être extraites de cette collinéation [6]. Il est aussi possible de déterminer le ratio ρ_j , propre à un couple de projetés, entre la coordonnées Z_j du point 3D et d [17] :

$$\rho_j = \frac{Z_j}{d} = \frac{1 + \mathbf{n}^T \mathbf{R}^T (\mathbf{t} / d)}{\mathbf{n}^T \mathbf{R}^T \mathbf{K}^{-1} \mathbf{p}} \quad (1)$$

Si l'on considère maintenant $N+1$ images d'une scène plane, l'étude des mises en correspondance entre chaque couple de vues successives i et $i+$ permet d'obtenir N homographies métriques : ${}^0 \mathbf{H}, {}^1 \mathbf{H}, \dots, {}^{N-1} \mathbf{H}$. Si l'on considère les vues i et $i+$, la collinéation associée est telle que :

$$m_{i,j} \propto {}^i \mathbf{H}_{i+} m_{i+,j}, \quad (2)$$

$m_{i,j}$ étant les coordonnées métriques du projeté dans le plan image i du point 3D P_j .

Connaissant la collinéation entre les images $i+$ et $i+$, il est possible de projeter dans l'image i des primitives extraites de l'image $i+$, par la relation :

$$\begin{aligned} m_{i,j} &\propto {}^i \mathbf{H}_{i+} {}^{i+} \mathbf{H}_{i+} m_{i+,j} \\ m_{i,j} &\propto {}^i \mathbf{H}_{i+} m_{i+,j} \end{aligned}$$

De manière générale, on obtient :

$$m_{i,j} \propto \prod_{l=i}^{k-1} {}^l \mathbf{H}_{l+} m_{k,j} = {}^i \mathbf{H}_k m_{k,j}, \quad (3)$$

avec $i \leq \dots \leq N$. Bien sûr, le projeté d'une primitive visible dans $_{k+1}$ n'est pas forcément visible dans l'image $_i$. Cependant, la reprojection permet à chaque instant du déplacement de savoir quelles sont les primitives définies par le chemin d'images qui sont visibles dans l'image courante afin de les suivre. De même, on peut déterminer quelles sont les primitives proches du domaine de visibilité, ceci afin de déplacer le robot pour qu'elles deviennent visibles.

La méthode que nous proposons consiste à faire rentrer dans le cadre de l'image courante des primitives qui ne sont pas encore visibles. À partir des primitives visibles à un instant donné et du chemin d'images, on peut déduire quelles sont les prochaines primitives à observer pour se rapprocher de la position désirée.

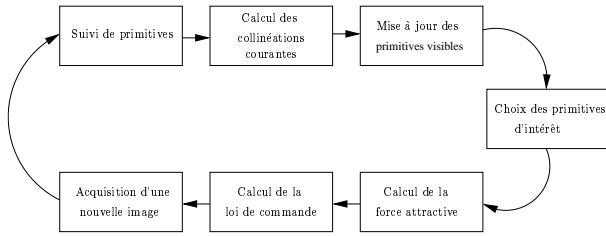


FIG. 4 – Boucle réalisée pour calculer le mouvement du robot.

La figure 4 présente les étapes que nous effectuons pour chaque image acquise par la caméra :

1. *Suivi des primitives visibles*, pour obtenir leur position dans la nouvelle image.
2. *Détermination des collinéations courantes*, entre la prise de vue courante et les images du chemin. Ces matrices permettent de connaître l'avancement du robot par rapport au chemin.
3. *Mise à jour des primitives visibles*: grâce aux collinéations précédentes, nous pouvons reprojeter l'ensemble des primitives dans le plan image courant, et alors mettre à jour l'ensemble des primitives visibles.
4. *Sélection des points d'intérêt*: cette étape consiste à choisir l'ensemble de points que l'on veut faire rentrer dans le champ de vision de la caméra, afin que le robot poursuive son chemin vers sa position désirée.
5. *Calcul de la force attractive*: à partir des positions courantes et du potentiel attractif que nous proposons, nous déduisons le sens de la force à appliquer au robot.
6. *Détermination de la loi de commande*: à partir de la force précédente, nous déduisons le mouvement que doit effectuer le robot.

Nous décrivons à présent ces différentes étapes, après la définition de quelques notations.

Notations. Soit $_{i+j}$ ($i \in [0, N]$), les $N + 1$ images du chemin. $_{0+}$ est l'image initiale obtenue avant le début du déplacement, et $_{N+}$ l'image désirée. On note \mathcal{M}_i l'ensemble des primitives mises en correspondance entre les images $_{i+}$ et $_{i+1}$. Un couple de projets de \mathcal{M}_j est représenté par $(m_{i,j}, m_{i+1,j})$. L'ensemble \mathcal{I}_i désigne l'ensemble des points 3D qui sont visibles dans l'image $_{i+}$ du chemin. On considère qu'un point $m_{t,j} \propto (u, v, 1)$ est visible si $u \in [u_m, u_M]$ et $v \in [v_m, v_M]$, où u_m, u_M, v_m et v_M forment un cadre dans l'image. On note \mathcal{C}_{free} cette zone de visibilité.

Dans une image $_{t+}$, les points considérés comme visibles sont notées $s_{t,j}$. V_i désigne l'ensemble des points de la vue $_{t+}$ qui sont visibles dans l'image $_{i+}$, soit :

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

Le système robotique est représenté par un vecteur \mathcal{X} dans l'espace des configurations \mathcal{W} . Le modèle 3D de la scène n'étant pas connu, nous choisissons comme dans [20] la paramétrisation partielle $\mathcal{X}_k = [{}^k\mathbf{t}_{d_N N}, \mathbf{u}\theta]$. À partir de la collinéation ${}^k\mathbf{H}_N$, il est possible de calculer ${}^k\mathbf{t}_{d_N N} = {}^k\mathbf{t}_N / d_N$, et ${}^k\mathbf{R}_N$. L'axe de rotation normalisé \mathbf{u} et l'angle de rotation θ sont déduits de la matrice ${}^k\mathbf{R}_N$.

Avant le début du déplacement du robot, nous calculons les collinéations ${}^i\mathbf{H}_{i+1}$ avec $i \in [0, N - 1]$ reliant les projets entre chaque couple d'images successives. L'ensemble des points du chemin est ensuite projeté sur le premier plan image $_{0+}$:

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

avec $j \in [1, N - 1]$ et les matrices ${}^0\mathbf{H}_k$ obtenues par composition des collinéations calculées précédemment grâce à la relation (3). L'ensemble des primitives visibles $s_{0,j}$ est initialisé en ne conservant seulement les projections qui appartiennent à \mathcal{C}_{free} . Les ensembles V_i sont initialisés en fonction des images d'origine de ces points visibles.

Nous allons maintenant présenter les traitements que nous effectuons sur chaque image prise par le caméra au cours de son déplacement.

Suivi des primitives. Dans l'image $_{t-1}$, on considère connaître la position d'un ensemble de primitives visibles $s_{t-1,j}$. Un suivi de primitives (nous utilisons la méthode de suivi de Shi-Tomasi-Kanade [24]) permet de connaître leur position $s_{t,j}$ dans l'image courante $_{t+}$. Certains points peuvent être sortis de la zone libre \mathcal{C}_{free} . Nous les conservons cependant pour déterminer les collinéations courantes (ces points sont hors de \mathcal{C}_{free} , mais encore dans le cadre de l'image courante). La phase 3 de mise à jour se chargera de les retirer. Les points qui sont rentrés dans le champ de vision de la caméra suite au mouvement du robot seront rajoutés dans l'ensemble des points visibles lors de la phase 3.

Détermination des collinéations courantes. Nous cherchons tout d'abord à déterminer les collinéations de la vue

courante vers l'ensemble des images i du chemin qui reste à parcourir. Deux possibilités se présentent :

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

où N_h est le nombre de points minimum pour calculer une homographie. Dans le premier cas, nous obtenons directement la collinéation en résolvant le système suivant :

$$s_{t,j} \propto {}^t\mathbf{H}_i m_{i,j}, \quad s_{t,j} \in V_i \quad (4)$$

Pour le deuxième cas, le contenu de l'image ne nous permet pas de calculer la collinéation par la méthode précédente. Si l'on considère la collinéation ${}^t\mathbf{H}_l$ entre la vue courante t et la vue l du chemin, obtenue par la relation (4), alors la collinéation entre la vue courante et la vue i du chemin est :

$${}^t\mathbf{H}_i = {}^t\mathbf{H}_l {}^l\mathbf{H}_i, \quad (5)$$

où $l < i < N$ et ${}^l\mathbf{H}_i$ est obtenue par composition des collinéations définies entre les images du chemin. Les images du chemin dont des primitives sont visibles dans t sont contiguës dans le chemin. Pour le calcul précédent, où l'on doit utiliser une collinéation calculée par la relation (4), nous choisirons celle qui relie la vue courante à l'image la plus avancée sur le chemin.

Les images d'origine des points visibles nous informent sur l'avancement du robot vis-à-vis du chemin initialement prévu. Il est donc possible de sélectionner les images dignes d'intérêt à un instant donné. Ainsi si l'image i est la première image du chemin dont des primitives sont visibles, il est inutile de calculer les collinéations vers des images antérieures. Le calcul (5) n'est donc effectué que pour les images en avant sur le chemin.

Mise à jour des points visibles. Grâce aux différentes collinéations obtenues, nous pouvons projeter sur le plan image courant les points $m_{i,j}$ depuis leurs images d'origine i . On obtient un ensemble de projetés $m_{t,j}$:

$$m_{t,j} \propto {}^t\mathbf{H}_i m_{i,j} \quad (6)$$

Cette projection n'est nécessaire que pour les primitives qui n'étaient pas encore visibles. En effet, les collinéations étant calculées à partir des primitives suivies ou par composition de collinéations, aucun gain de précision pour le suivi n'est envisageable en reprojetant une primitive déjà visible.

De plus, une primitive est nécessairement observable depuis au moins deux images du chemin. Deux résultats de projection sont alors possibles pour ce même point. Pour limiter les erreurs de reprojection, nous utiliserons comme référence la première image où le point apparaît (car cette image est la plus proche).

Les points $s_{t,j}$ suivis depuis l'image $t-$ sont filtrés pour retirer ceux qui sont sortis de la zone libre \mathcal{C}_{free} . On rajoute ensuite les points obtenus par la projection (6) qui sont tels que $m_{t,j} \in \mathcal{C}_{free}$. Les ensembles V_i peuvent ensuite être mis à jour, à partir du nouvel ensemble $s_{t,j}$.

Sélection de l'ensemble de points d'intérêt. Nous cherchons ici l'indice i^* permettant, parmi tous les ensembles d'appariements \mathcal{M}_i définis sur le chemin, de sélectionner l'ensemble \mathcal{M}_{i^*} grâce auquel nous allons calculer le mouvement du robot. Les projections des points de l'ensemble d'intérêt doivent être proche de la zone \mathcal{C}_{free} . On va donc considérer un ensemble dont une partie des primitives est déjà visible par la caméra. En effet, si des points de l'ensemble \mathcal{M}_i sont actuellement dans le champ de vision de la caméra, du point de vue du chemin à parcourir, la caméra se situe entre l'image i et i_+ . En déplaçant le robot pour observer l'ensemble des primitives définies dans \mathcal{M}_i , le champ de vision de la caméra va se rapprocher de celui associé à l'image i_+ . Des primitives de l'image i_+ appartenant à l'ensemble d'appariement \mathcal{M}_{i+} vont alors devenir visibles. Une fois que suffisamment de primitives de l'ensemble \mathcal{M}_{i+} seront visibles, nous pourrons alors considérer cet ensemble comme celui d'intérêt.

Nous choisissons donc, parmi tous les ensembles vérifiant

$ard(m_{i^*,j} | m_{i^*,j} \in V_{i^*} \quad (m_{i^*,j}, m_{i^*+1,j}) \in \mathcal{M}_{i^*}) > N_M$ celui de rang i^* maximal (N_M étant un seuil). Nous montrons ensuite comment à partir de cet ensemble \mathcal{M}_{i^*} et de la fonction de potentiel que nous proposons, nous calculons la force attractive à appliquer au robot.

Détermination de la force attractive. La force attractive est définie par [19] :

$$\mathbf{F}_f(\mathbf{x}) = -\epsilon \left(\frac{\partial \mathbf{f}}{\partial \mathcal{X}} \right)^+ \vec{\mathbf{f}}^T \mathbf{f},$$

où \mathbf{f} est une fonction dérivable sur tout l'espace des configurations \mathcal{W} et $\vec{\mathbf{f}}^T \mathbf{f}$ le gradient d'une fonction de potentiel $\mathbf{f} = (\mathbf{f}(\mathcal{X}))$. ϵ est un gain positif réglant l'amplitude de la force.

Le potentiel attractif \mathbf{f} est défini dans le plan image, afin d'attirer dans le champ de vision des primitives qui ne sont pas encore visibles. Nous proposons le potentiel suivant :

$$s = \sum_j s(s_j), \quad (7)$$

avec :

$$s(s_j) = g(u_j - u_M) + g(v_j - v_M) + g(u_m - u_j) + g(v_m - v_j),$$

et

$$g(x) = x * \pi^- \arctan(\pi x) + t \quad \mathbf{a}$$

où s_j est le vecteur des coordonnées $(u_j \ v_j \ 1)^T$ de la primitive que l'on veut faire rentrer dans le cadre de l'image. t et t sont des constantes. Ce potentiel est nul lorsque toutes les primitives sont dans \mathcal{C}_{free} . La figure 5 représente cette fonction de potentiel dans le cas d'un seul point.

La force attractive résultante est alors :

$$\mathbf{F}_s = -\epsilon \left(\frac{\partial \mathbf{s}}{\partial \mathcal{X}} \right)^+ \vec{\mathbf{s}}^T \mathbf{s} = -\epsilon \mathbf{L}^+ \vec{\mathbf{s}}^T \mathbf{s},$$

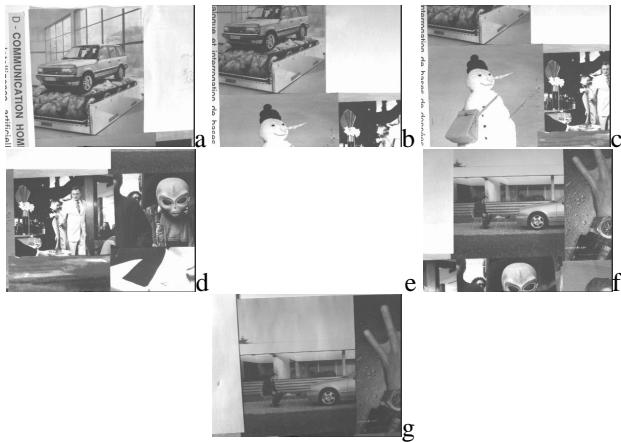


FIG. 6 – Images constituant le chemin à parcourir (a : position initiale, g : position finale)

La figure 7 représente une reprojection dans le plan de l'image initiale de l'ensemble des points d'intérêts. Les cadres images des autres vues du chemin sont aussi projetés sur ce plan. On peut noter qu'un grand nombre de points n'est pas visible dans la première image.

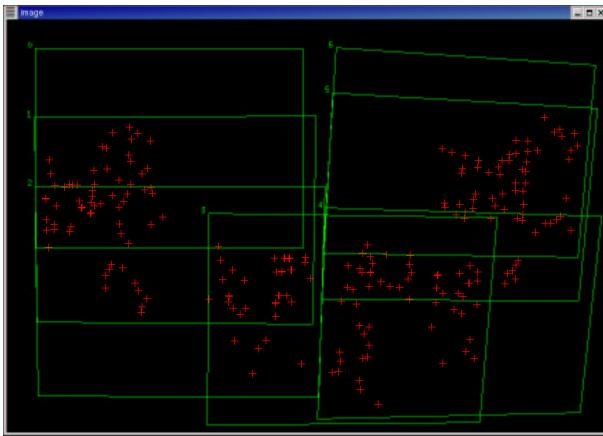


FIG. 7 – Reprojection de tous les points d'intérêt et plans images sur le 1^{er} plan image

La figure 8 présente la trajectoire suivie par le point principal de la caméra lors du déplacement. On peut noter que le système robotique ne cherche pas à atteindre les positions intermédiaires associées aux images du chemin. Les trajectoires 3D et du robot sont comparées à deux autres méthodes sur les figures 9 et 10. Dans la première méthode [20], une planification basée sur une décomposition temporelle des matrices de collinéation entre chaque couple d'images fournit une trajectoire que le système robotique suit ensuite par asservissement visuel . Des différentes images du chemin correspondent à des positions qu'atteint successivement le robot. Dans la deuxième méthode, le

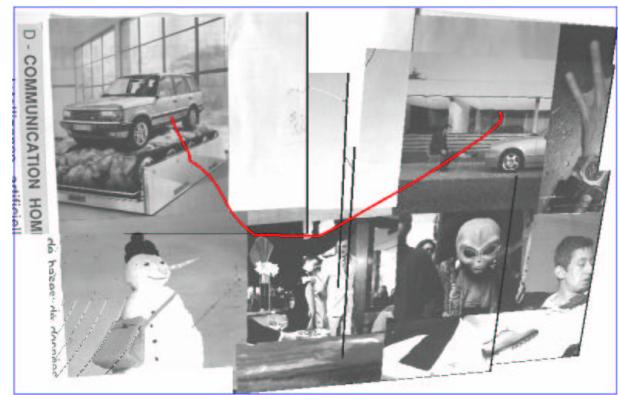


FIG. 8 – Trajectoire du point principal reprojeté dans le premier plan image

système robotique converge vers les positions intermédiaires par asservissement visuel . Mais, à chaque itération de la boucle de commande, les primitives en commun entre l'image cible courante et l'image suivante sur le chemin sont projetées sur le plan image courant (reprojection par la matrice de collinéation). Si suffisamment de points sont visibles dans le cadre de l'image, alors le système s'asservit vers l'image suivante du chemin. Le robot n'atteint donc pas les positions associées aux images du chemin (comme on le voit sur le schéma 10), mais il est tout de même dépendant des attitudes intermédiaires.

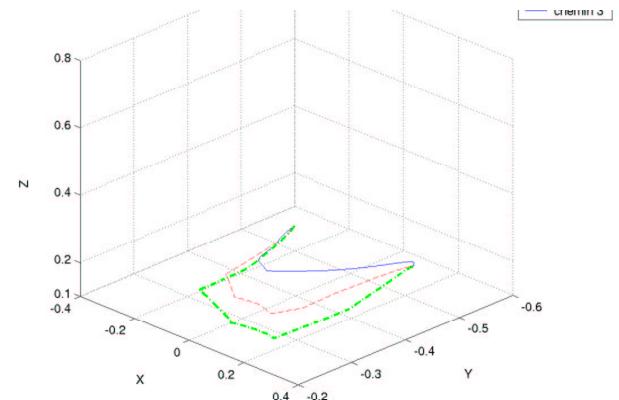


FIG. 9 – Trajectoire 3d réalisée par le robot pour le chemin de la figure 6

La méthode que nous présentons dans cet article correspond à une trajectoire encore plus courte, tout en satisfaisant la contrainte de visibilité. Le fait de se déplacer pour faire rentrer des primitives dans l'image ne pénalise donc aucunement le déplacement du robot, bien au contraire. La section suivante démontre l'indépendance totale de notre méthode vis-à-vis des positions associées aux images intermédiaires.

- 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] C. Froidevaux, M. Gaudel, et M. Soria. *Types de données et algorithmes*. Mc Graw-Hill, 1992.
 - [9] S. K. Ghosh et D. M. Mount. An output sensitive algorithm for computing visibility graphs. In *IEEE Symp. on Foundations of Computer Science, Los Angeles*, 1987.
 - [10] P. Gros, A. Remazeilles, et F. Chaumette. Les invariants : introduction et application à la vision et à la robotique. In *Actes des troisièmes journées nationales de la recherche en robotique*, Giens, France, oct. 2001.
 - [11] H. Haddad, M. Khatib, S. Lacroix, et R. Chatila. Reactive navigation in outdoor environments using potential fields. In *IEEE Int. Conf. on Robotics and Automation*, pages 1232–1237, Louvain (Belgique), mai 1998.
 - [12] C. Harris et M. Stephens. A combined corner and edge detector. In *Alvey Vision Conf.*, pages 147–151, University of Manchester, England, sept. 1988.
 - [13] S. Hutchinson, G.D. Hager, et P.I. Corke. A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5):651–670, oct. 1996.
 - [14] S. Jones, C. Andersen, et J. L. Crowley. Appearance based processes for visual navigation. *IEE/RSJ Int. Conf. on Intelligent Robots and Systems*, sept. 1997.
 - [15] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–98, 1986.
 - [16] E. Malis et 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, avril 2002.
 - [17] E. Malis, F. Chaumette, et S. Bouvet. 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, avril 1999.
 - [18] Y. Matsumoto, M. Inaba, et H. Inoue. Visual navigation using view-sequenced route representation. In *Int. Conf. on Robotics and Automation*, pages 83–88, Minneapolis, 1996.
 - [19] Y. Mezouar. *Planification de trajectoires pour l'asservissement visuel*. Thèse de doctorat, Université de Rennes I, 2001.
 - [20] Y. Mezouar, A. Remazeilles, P. Gros, et 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, mai 2002.
 - [21] C. Nissoux, T. Siméon, et J-P. Laumond. Visibility based probabilistic roadmaps. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1316–1321, Kyongju, Corée, oct. 1999.
 - [22] C. Schmid et R. Mohr. Local grayvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence*, 19(5):530–534, May 1997.
 - [23] A. Shashua et 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.
 - [24] J. Shi et C. Tomasi. Good features to track. In *IEEE Computer Vision and Pattern Recognition*, pages 593–600, Seattle, juin 1994.
 - [25] A.C. Victorino, P. Rives, et 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.
 - [26] J. Wolf, W. Burgard, et 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, mai 2002.
 - [27] 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.
 - [28] Z. Zhang, R. Deriche, Q. Luong, et O. Faugeras. A robust approach to image matching : Recovery of the epipolar geometry. *Int. Symp. of Young Investigators on Information-Computer-Control*, 1994.
 - [29] D. Zhu et J.C. Latombe. New heuristic algorithms for efficient hierarchical path planning. *IEEE Trans. on Robotics and Automation*, 7:9–20, 1991.