

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

Résumé

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.

Mots Clef

vision robotique, reconnaissance d'images, champs de potentiel

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

Keywords

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 Voronoï [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 repose sur une phase préalable de planification, c'est-à-dire que l'ensemble de la trajectoire est calculée 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 précisées. Les techniques appelées *SLAM* (*Simultaneous Localisation 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 se limitent actuellement à des espaces d'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 définis sur le voisinage de chaque primitive des images. Bien qu'une caméra soit utilisée, les scènes traitées se limitent aux environnements structurés, et aucune 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 (déplacements entre les cellules connexes 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 à partir 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 la série d'images. La section 3 détaille ensuite comment la méthode des champs de potentiels peut être adaptée pour calculer le mouvement du robot. Enfin la section 4 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 acquise, les images qu'elle contient sont traitées afin de générer un graphe valué en fonction des mises 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 des vues initiale et désirée par un algorithme d'indexation par le contenu. À ce niveau, les images initiales et désirées forment donc des images requête à la base,
- appariement des images les plus proches avec les deux images requêtes afin de compléter le graphe de l'application,
- détermination de la série d'images par recherche du plus court chemin dans le graphe.

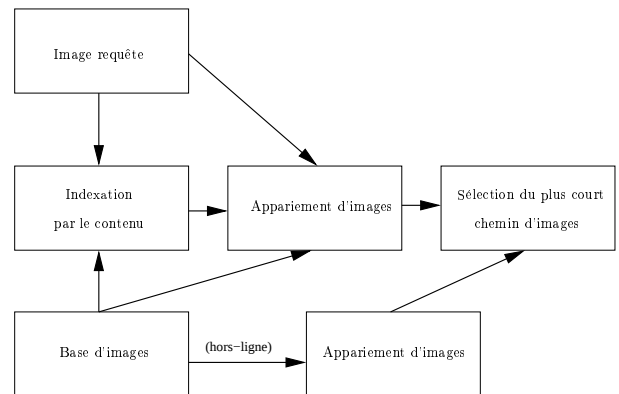


FIG. 1 – Opérations pour la détermination du chemin d'images

La section suivante présente la construction du graphe ainsi que l'algorithme d'appariement que nous utilisons. La section 2.2 détaille la phase d'indexation, et la section 2.3 montre comment est obtenue la série d'images.

2.1 Construction du graphe des images de la base

Pour déterminer quelles sont les images qui caractérisent le mieux l'environnement que doit traverser le système robotique, nous nous basons sur les correspondances entre les images de la base. Celle-ci étant acquise hors-ligne, nous pouvons nous permettre d'effectuer un appariement robuste. Nous utilisons pour cela l'algorithme de Zhang [28], qui élimine les mauvais appariements par la détermination de la géométrie épipolaire. Les primitives mises en correspondance sont des points d'intérêt correspondant à de fortes courbures du signal en niveau de gris [12].

A partir de ces mises en correspondance, nous réalisons un graphe valué :

- à chaque image de la base est associé un nœud dans le graphe,
- les nœuds sont reliés par un arc dont la valuation correspond à la facilité de se déplacer d'une image vers l'autre image.

La valuation est définie comme l'inverse du nombre de points appariés. En effet, plus les images ont de points en correspondance, plus les deux positions associées sont proches et donc le déplacement aisé.

2.2 Reconnaissance d'une image requête

On rappelle tout d'abord que deux requêtes sont successivement traitées : l'une pour l'image désirée, et l'autre pour l'image initiale.

Une possibilité pour déterminer les images les plus proches d'une image requête serait d'apparier celle-ci avec toutes les images de la base. Mais cela aboutirait à des temps de calcul beaucoup trop importants. Nous allons donc rechercher les images les plus proches en terme de contenu par une technique d'indexation.

Pour ce faire, des descripteurs sont calculés hors-ligne pour tous les points détectés dans les images de la base. Nous utilisons les invariants photométriques, grandeurs invariantes aux translations, rotations, changements d'échelle, et variations de l'illumination [22, 10]. Les mêmes vecteurs caractéristiques sont déterminés pour les points extraits de l'image requête.

La reconnaissance s'effectue ensuite par une recherche des k -Plus Proches Voisins [1] : chaque descripteur de l'image requête donne une voix aux k descripteurs de la base qui lui sont les plus proches (la distance euclidienne est utilisée), et les images les plus proches sont celles qui possèdent au final le plus de voix.

Le résultat d'une recherche dans la base est présenté sur la figure 2. Notre base contient actuellement une cinquantaine d'images. De chaque image est extraite une centaine de points d'intérêt. L'image recherchée n'était pas contenue dans la base considérée.

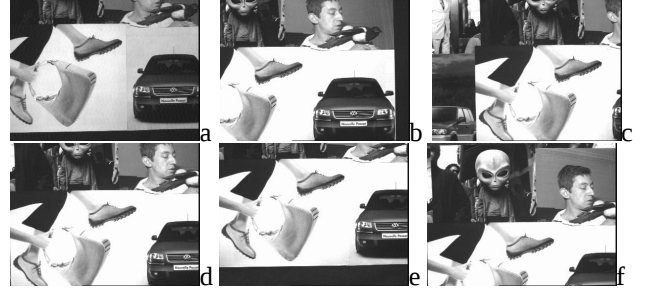


FIG. 2 – Résultats de l'indexation pour l'image requête *a*. Les scores obtenus sont : *b* : 57, *c* : 54, *d* : 52, *e* : 50, *f* : 47

2.3 Détermination du chemin global

Connaissant les images les plus proches des images initiale et désirée, nous pouvons alors rajouter leurs nœuds dans le graphe obtenu sur la base. Pour ce faire, nous les apparions avec leurs images les plus proches respectives. De ces appariements sont déduites les valeurs associées aux arcs vers leurs nœuds d'entrées dans le graphe.

L'algorithme du plus court chemin de Dijkstra [8] permet alors de déterminer les images de la base reliant les deux images requêtes. Cette méthodologie permet d'assurer que :

- chaque couple d'images successives du chemin possède une zone de recouvrement,
- le chemin choisi est le plus court pour la valuation des arcs choisie.

La figure 3 présente un exemple de chemin d'images extraites de la base. On note que les couples d'images successives possèdent bien une zone de recouvrement.

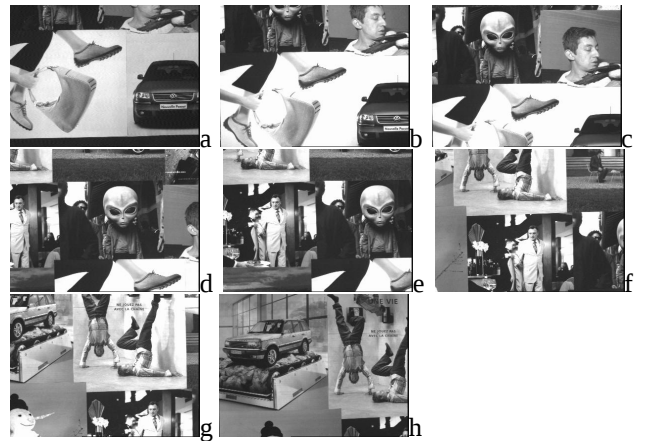


FIG. 3 – Exemple de chemin d'images. *a* est l'image initiale, et *h* l'image désirée. Les autres images ont été extraites automatiquement de la base.

La section suivante présente la méthode que nous proposons pour définir les mouvements du robot.

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 2D), ce 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 convergait 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 reprojections 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.

3.1 Géométrie de N images

Nous considérons tout d'abord deux prises de vue ψ_1 et ψ_2 d'une même scène. Un plan de référence Π est représenté dans le repère de la deuxième prise de vue $\mathcal{F}_2$ par le vecteur $\pi^T = [\mathbf{n}_2 \quad -d_2]$, où $\mathbf{n}_2$ est son vecteur unitaire et d_2 la distance orthogonale du plan au centre optique de la deuxième caméra. Si l'on considère une scène plane, alors les coor-

données pixeliques des projections sur les deux images, $p_1 = [u_1 \ v_1 \ 1]$ et $p_2 = [u_2 \ v_2 \ 1]$, d'un point de la scène $P \in \Pi$ sont liées par la transformation projective [6] :

$$p_1 \propto \mathbf{G}_\pi p_2,$$

où $\mathbf{G}_\pi$ est la collinéation en pixel, relative au plan Π . Elle correspond à un changement de repère projectif de l'image ψ_2 vers ψ_1 ($\propto$ 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 collinéaires 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 :

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

(${}^1\mathbf{R}_2 \ {}^1\mathbf{t}_2$) représentant le changement de repère entre les deux prises de vue $\mathcal{F}_1$ et $\mathcal{F}_2$. 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_2 [17] :

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

Si l'on considère maintenant $N + 1$ images d'une scène plane, l'étude des mises en correspondance entre chaque couple de vues successives ψ_i et ψ_{i+1} permet d'obtenir N homographies métriques : ${}^0\mathbf{H}_1, \dots, {}^{N-1}\mathbf{H}_N$. Si l'on considère les vues ψ_i et ψ_{i+1} , la collinéation associée est telle que :

$$m_{i,j} \propto {}^i\mathbf{H}_{i+1} m_{i+1,j}, \quad (2)$$

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

Connaissant la collinéation entre les images ψ_{i+1} et ψ_{i+2} , il est possible de reprojetter dans l'image ψ_i des primitives extraites de l'image ψ_{i+2} , par la relation :

$$\begin{aligned} m_{i,j} &\propto {}^i\mathbf{H}_{i+1} {}^{i+1}\mathbf{H}_{i+2} m_{i+2,j} \\ m_{i,j} &\propto {}^i\mathbf{H}_{i+2} m_{i+2,j} \end{aligned}$$

De manière générale, on obtient :

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

avec $i \leq k \leq N$. Bien sûr, le projeté d'une primitive visible dans ψ_k n'est pas forcément visible dans l'image ψ_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.

3.2 Méthodologie proposée

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.

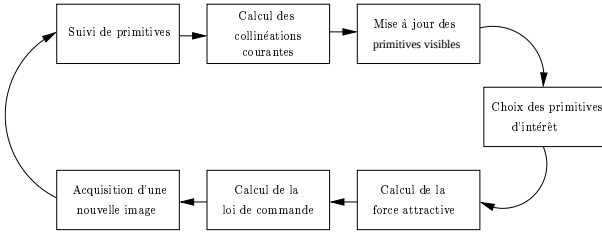


FIG. 4 – Boucle réalisée pour calculer le mouvement du robot.

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 ψ_i ($i \in [0, N]$), les $N + 1$ images du chemin. ψ_0 est l'image initiale obtenue avant le début du déplacement, et ψ_N l'image désirée. On note $\mathcal{M}_i$ l'ensemble des primitives mises en correspondance entre les images ψ_i et ψ_{i+1} . Un couple de projetés 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 ψ_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 ψ_t , les points considérés comme visibles sont notées $s_{t,j}$. V_i désigne l'ensemble des points de la vue ψ_t qui sont visibles dans l'image ψ_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 ${}^t\mathbf{R}_N$. L'axe de rotation normalisé $\mathbf{u}$ et l'angle de rotation θ sont déduits de la matrice ${}^t\mathbf{R}_N$.

Avant le début du déplacement du robot, nous calculons les collinéations ${}^i\mathbf{H}_{i+1}$ avec $i \in [0, N - 1]$ reliant les projetés entre chaque couple d'images successives. L'ensemble des points du chemin est ensuite projeté sur le premier plan image ψ_0 :

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

avec $k \in [1, N - 1]$ 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 ψ_{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 ψ_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 ψ_i du chemin qui reste à parcourir. Deux possibilités se présentent :

- $\text{card}(V_i) \geq N_h$
- $\text{card}(V_i) < N_h$,

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 :

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

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 ψ_t et la vue ψ_l du chemin, obtenue par la relation (4), alors la collinéation entre la vue courante et la vue ψ_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 ψ_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 ψ_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 ψ_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 reprojectant 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 reprojectation, 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 ψ_{t-1} 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 ψ_i et ψ_{i+1} . 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 ψ_{i+1} . Des primitives de l'image ψ_{i+1} appartenant à l'ensemble d'appariement $\mathcal{M}_{i+1}$ vont alors devenir visibles. Une fois que suffisamment de primitives de l'ensemble $\mathcal{M}_{i+1}$ seront visibles, nous pourrons alors considérer cet ensemble comme celui d'intérêt.

Nous choisissons donc, parmi tous les ensembles vérifiant

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

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] :

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

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

Le potentiel attractif $\mathcal{V}_f$ 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 :

$$\mathcal{V}_s = \sum_j \mathcal{V}_s(s_j), \quad (7)$$

avec :

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

et

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

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. k 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 :

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

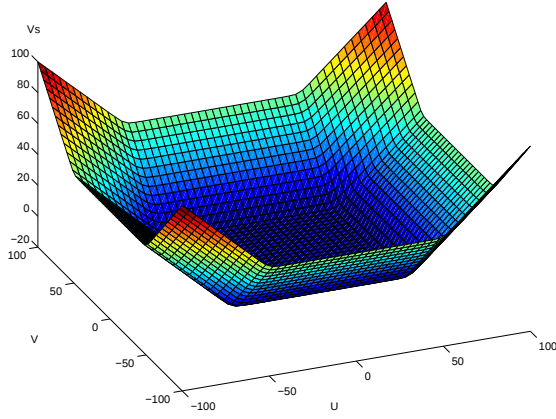


FIG. 5 – Fonction de potentiel associée à la contrainte de visibilité

où $\vec{\nabla}_s^T \mathcal{V}_s$ est aisément obtenu à partir de (7) et $\mathbf{L}$ est la matrice d'interaction associée à s [5]. Elle définit le mouvement de primitives visuelles en fonction de la vitesse de la caméra $\mathcal{T}_c$: $\dot{s} = \mathbf{L}\mathcal{T}_c$. Pour un point P de coordonnées $(X \ Y \ Z)$ dans le repère caméra et dont la projection est $\mathbf{p} = (x \ y)$ en coordonnées métriques, et pour une focale de un mètre, elle est donnée par :

$$\mathbf{L}(\mathbf{p}, Z) = \begin{pmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix}$$

Pour un ensemble de k points, elle devient :

$$\mathbf{L}(s, \mathbf{Z}) = [\mathbf{L}^T(\mathbf{p}_1, Z_1) \ \dots \ \mathbf{L}^T(\mathbf{p}_k, Z_k)]^T \quad (8)$$

En combinant la matrice d'interaction pour un point et la relation (1), on obtient :

$$\mathbf{L}(\mathbf{p}, d_{i^*+1}) = \frac{1}{d_{i^*+1}} [\mathbf{S} \ \mathbf{Q}] ,$$

où $\mathbf{Q} = [\mathbf{Q}_1^T \dots \mathbf{Q}_k^T]$ et $\mathbf{S} = [\mathbf{S}_1^T \dots \mathbf{S}_k^T]$ sont deux matrices $2n \times 3$ indépendantes de d_{i^*+1} . Les matrices $\mathbf{S}_j$ et $\mathbf{Q}_j$ sont de la forme :

$$\mathbf{S}_j = \begin{bmatrix} -\frac{1}{\rho_{t,j}} & 0 & \frac{x}{\rho_{t,j}} \\ 0 & -\frac{1}{\rho_{t,j}} & \frac{y}{\rho_{t,j}} \end{bmatrix}$$

$$\mathbf{Q}_j = \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix}$$

Les ratios $\rho_{t,j}$ sont déduits de la collinéation ${}^t\mathbf{H}_{i^*+1}$ entre la vue courante et la vue ψ_{i^*+1} :

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

où p_j désigne les coordonnées dans l'image courante du point P_j considéré ($p_j = s_{t,j}$ si le point est visible).

Loi de commande. À partir de la force calculée par la méthode précédente, le robot est directement asservi en vitesse pour aller dans la direction de cette force :

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

Une nouvelle image ψ_{t+1} est ensuite acquise. Ce processus boucle jusqu'à ce que suffisamment de primitives P_j visibles dans l'image désirée ψ_N soient dans le champ de vision de la caméra. Nous pouvons alors considérer que le système robotique est dans le voisinage de la position désirée. Le robot peut donc à ce moment se diriger vers la position désirée.

Convergence vers la position désirée. Lorsque la position désirée est proche, nous utilisons la force attractive classique :

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

où le potentiel est représenté comme une fonction parabolique,

$$\mathcal{V}_a = \frac{1}{2} \|\mathcal{X}_t - \mathcal{X}^*\|^2$$

Sachant que $\mathcal{X}^* = \mathbf{0}_{6 \times 1}$, la force attractive est :

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

Le mouvement du robot est déduit de $\mathcal{T}_c = \mathbf{F}_a$. Cette loi de commande correspond exactement à un asservissement visuel hybride [16]. Il serait aussi possible d'utiliser un asservissement visuel 2D, basée sur la position courante des primitives et leur position désirée dans l'image ψ_N .

Dans la nouvelle image obtenue, les primitives visibles de l'image ψ_N permettent de calculer la collinéation ${}^t\mathbf{H}_N$. Par projection, les primitives précédemment non visibles de ψ_N sont rajoutées à $s_{t,j}$ si elles rentrent dans le champ de vision de la caméra. De la collinéation précédente nous déterminons la rotation ${}^t\mathbf{R}_N$ et la translation à un facteur d'échelle près ${}^t\mathbf{t}_{d_N}$, qui nous permettent de mettre à jour la position de la caméra $\mathcal{X}_t$ et donc la force à appliquer. Une nouvelle image est acquise, et ce processus est répété jusqu'à ce que le robot atteigne sa position désirée.

La partie suivante présente des résultats expérimentaux qui valident notre approche.

4 Résultats expérimentaux

Les expériences que nous présentons ont été réalisées sur un bras articulé à six degrés de liberté, doté d'une caméra montée sur son effecteur. L'environnement de déplacement est constitué d'un plan sur lequel nous avons posé une série de photos diverses. Pour montrer la validité de notre approche, nous avons choisi un déplacement où le robot ne peut aller directement en ligne droite de la position initiale à la position finale. Les images définissant le parcours à réaliser, extraites de la base comme indiqué dans la section 2 sont présentées sur la figure 6. Le mouvement correspond à un déplacement en "U" (voir la figure 8).

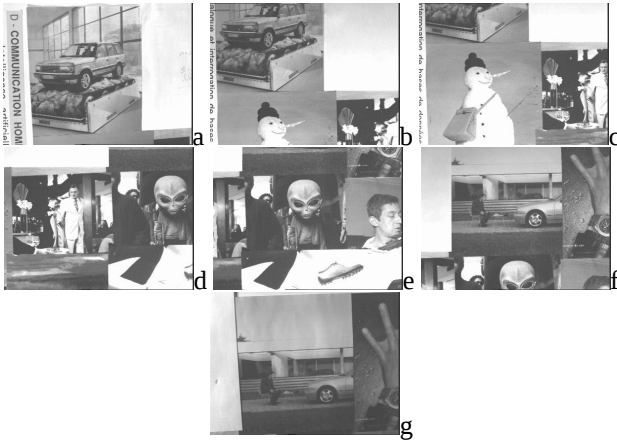


FIG. 6 – Images constituant le chemin à parcourir (a : position initiale, g : position finale)

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.

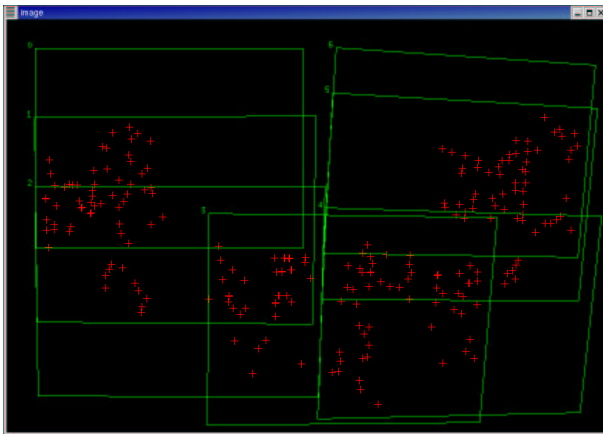


FIG. 7 – Reprojection de tous les points d'intérêt et plans images sur le 1^{er} plan image

4.1 Comparaison de la trajectoire obtenue

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 2D 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 2D. Les différentes images du chemin correspondent à des positions qu'atteint successivement le robot. Dans la deuxième méthode, le

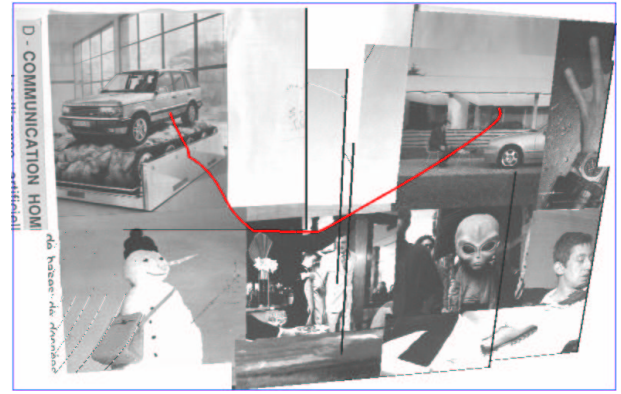


FIG. 8 – Trajectoire du point principal reprojecté dans le premier plan image

système robotique converge vers les positions intermédiaires par asservissement visuel 2D. 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.

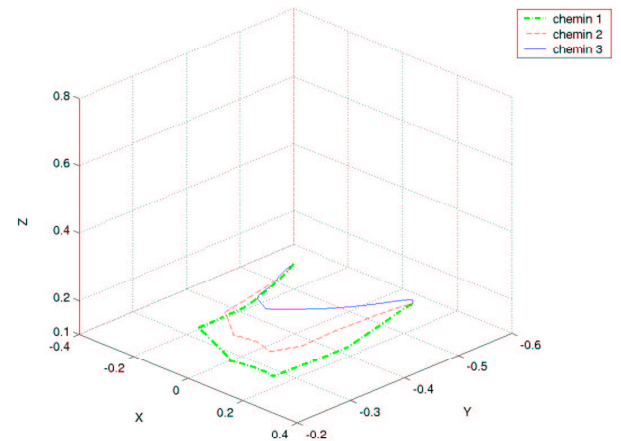


FIG. 9 – Trajectoire 3d réalisée par le robot pour le chemin de la figure 6

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.

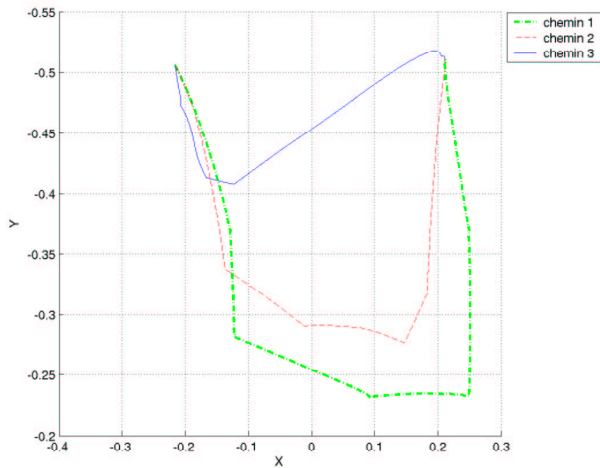


FIG. 10 – Trajectoire 2d réalisée par le robot pour le chemin de la figure 6

4.2 Robustesse à la position des images intermédiaires

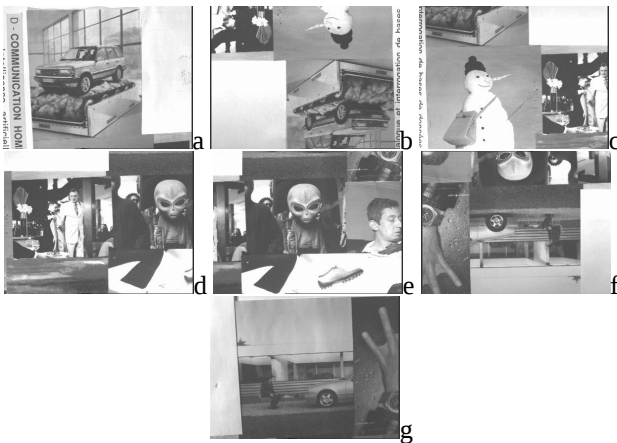


FIG. 11 – Images constituant le chemin à parcourir, avec deux images inversées (a : position initiale, g : position finale)

Pour montrer l'indépendance aux positions des images intermédiaires, nous avons appliqué une rotation de 180 degrés aux images *b* et *f* du chemin (le chemin est alors celui de la figure 11). La résolution de ce chemin par [20] contraindrait le robot à effectuer des rotations inutiles lors des déplacements *ab*, *bc*, *ef* et *fg*. La deuxième méthode, bien qu'elle permette de ne pas converger vers les images intermédiaires, réaliserait également une partie de ces rotations. La figure 12 compare la trajectoire associée au chemin sans rotation présenté en 6, avec la trajectoire que l'on obtient lorsque les images *b* et *f* sont retournées. Les deux trajectoires sont quasiment identiques, ce qui prouve bien que notre méthode est indépendante des positions des prises de vue.

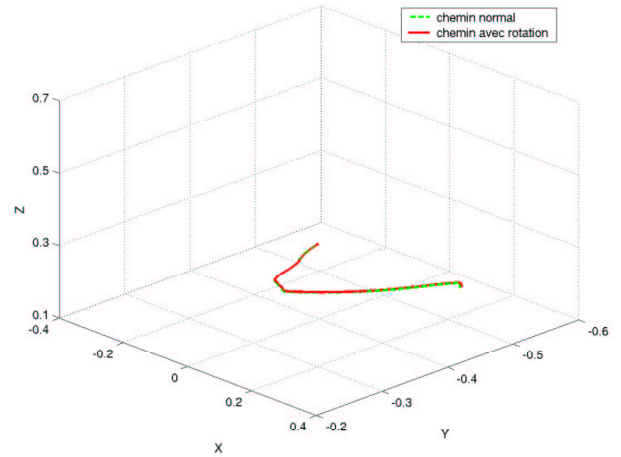


FIG. 12 – Comparaison de la trajectoire du robot réalisée pour les chemins des figures 6 et 11

5 Conclusion

Nous avons présenté dans cet article une nouvelle méthode pour la définition des déplacements d'un robot à l'aide d'informations visuelles. Le chemin à parcourir est défini comme une succession d'images extraites d'une base de l'environnement. Notre méthode, qui utilise le formalisme des champs de potentiel, évite les minima locaux par l'emploi d'un seul potentiel attractif visant à faire rentrer dans le champ de vision de la caméra les prochaines primitives d'intérêt définies sur le chemin. Nous avons de plus montré que la trajectoire réalisée par le robot ne dépend pas des positions associées aux images du chemin. Cependant, notre cadre applicatif se limite actuellement aux environnements plans. Nous travaillons actuellement sur ce point, et envisageons aussi d'intégrer des contraintes non holonomes pour effectuer des expérimentations avec un véhicule robotisé.

Références

- [1] S.A. Berrani, L. Amsaleg, et 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] F. Chaumette. Asservissement visuel. In W. Khalil, editor, *La commande des robots manipulateurs*, Traité IC2, chapter 3, pages 105–150. Hermès, 2002.
- [4] J. L. Crowley. Navigation for intelligent mobile robot. *IEEE Journal of Robotics and Automation*, 1(1), 1985.
- [5] B. Espiau, F. Chaumette, et 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 et 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] 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. *IEEE/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. Boudet. 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 and 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.