

Asservissement visuel qualitatif pour la navigation

Qualitative visual servoing for navigation

Anthony Remazeilles, François Chaumette, Patrick Gros

¹IRISA, Campus de Beaulieu 35042 Rennes Cedex, France
Prénom.Nom@irisa.fr

Manuscrit reçu le 12 janvier 2006

Résumé et mots clés

Dans cet article, une nouvelle méthode est proposée pour contrôler les mouvements d'un système robotique à l'aide d'un capteur de vision monoculaire durant une tâche de navigation. Cette approche s'appuie sur une représentation topologique de l'environnement, où la scène est directement décrite dans l'espace du capteur par une base d'images acquises hors-ligne. Lors de la navigation, une étape préalable de recherche d'images permet de localiser la position courante du robot, en mettant en relation la vue que sa caméra fournit avec celles stockées dans la base. Un algorithme classique de recherche de plus-court chemin permet alors d'extraire de la base un ensemble de vues caractérisant l'espace à parcourir afin de rejoindre la position désirée. Cet article se concentre principalement sur la loi de commande permettant de déduire les mouvements du robot en fonction des informations extraites de ce chemin et de la vue courante de la caméra. Notre méthode ne s'appuie pas sur un modèle 3D de la scène, et n'effectue pas une planification temporelle de la trajectoire à réaliser. De plus, les images du chemin ne sont pas considérées comme des positions désirées intermédiaires vers lesquelles doit converger la caméra. Le schéma d'asservissement visuel proposé, qualifié de qualitatif, repose sur des fonctions de coût, et assure que le robot peut toujours observer les amers visuels initialement détectés sur le chemin d'images. Des expériences réalisées en simulation et avec un système réel montrent que le formalisme proposé permet de contrôler les mouvements d'une caméra dans un environnement 3D.

Robotique, asservissement visuel, vision par ordinateur.

Abstract and key words

We propose in this article a novel approach for vision-based control of a robotic system during a navigation task. This technique is based on a topological representation of the environment in which the scene is directly described within the sensor space, by an image database acquired off-line. Before each navigation task, a preliminary step consists in localizing the current position of the robotic system. This is realized through an image retrieval scheme, by searching within the database the views that are the most similar to the one given by the camera. Then a classical shortest path finding algorithm enables to extract from the database a sequence of views that visually describe the environment the robot has to go through in order to reach the desired position. This article mainly focuses on the control law that is used for controlling the motions of the robotic system, by comparing the visual information extracted from the current view and from the image path. This control law does not need a CAD model of the environment, and does not perform a temporal path planning. Furthermore, the images from the path are not considered as successive desired positions that have to be consecutively reached by the camera. The qualitative visual servoing scheme proposed, based on cost functions, ensures that the robotic system is always able to observe some visual features initially detected on the image path. Experiments realized in simulation and with a real system demonstrate that this formalism enables to control a camera moving in a 3D environment.

Robotics, visual servoing, computer vision.

1. Introduction

Les tâches de navigation diffèrent des tâches de positionnement de part la taille de l'environnement que doit traverser le système robotique. Dans le premier cas, les positions initiale et désirée peuvent être situées dans des zones totalement différentes. Du point de vue des capteurs de vision, cette particularité se traduit par l'impossibilité de mettre en correspondance les informations visuelles obtenues depuis la position initiale avec celles associées à la position finale. Naturellement, l'absence de recouvrement entre les deux images rend un déplacement autonome impossible entre ces deux positions sans connaissance supplémentaire.

La réalisation de tels déplacements impose de donner au système robotique une représentation interne de son environnement de navigation. Cette connaissance est utilisée pour localiser la position courante du système, ainsi que la position qu'il doit atteindre. Elle permet de plus de définir un chemin reliant ces deux positions, et fournit toutes les informations nécessaires à la réalisation proprement dite du déplacement.

De nombreux types de représentation ont été proposés dans la littérature. Les deux grandes familles qui s'en dégagent sont les méthodes basées modèle et les approches dites topologiques. Elles sont présentées succinctement ci-dessous.

1.1. Approche basée modèle

L'approche basée modèle repose sur la connaissance du modèle 3D de la scène. La localisation du système robotique est réalisée en recalant le modèle local, déduit des informations capteurs, avec le modèle global qui est exprimé dans un repère de référence. De nombreux types de primitives ont été considérées dans la littérature. On citera par exemple les lignes [Dao *et al.*, 2003], les plans [Cobzas *et al.*, 2003] ou encore les points [Burschka and Hager, 2001, Royer *et al.*, 2004]. Dans le cas où le modèle n'est pas connu *a priori*, celui-ci peut être obtenu lors d'une phase préalable de reconstruction. Durant cette phase, le robot est généralement contrôlé par un opérateur humain. Dans [Royer *et al.*, 2004], le modèle de l'environnement est obtenu par un ajustement hiérarchique de rayons. Dans [Burschka and Hager, 2001], un système de suivi est couplé avec un odomètre afin d'estimer les coordonnées sphériques de points. Notons de plus que de nombreux travaux s'intéressent à la reconstruction automatique de l'environnement, durant une phase d'exploration où le robot est autonome (approche dite du SLAM, pour *Simultaneous Localization And Mapping* [Thrun *et al.*, 2000, Se *et al.*, 2002]). Cependant, les mouvements réalisés ont pour but de découvrir de nouvelles régions de l'espace et ne permettent pas au robot de se rendre vers une position particulière de l'environnement.

Dans les approches basées modèle, une fois que la pose du robot a été estimée, la navigation est généralement réalisée en se définissant un ensemble de buts intermédiaires.

Ainsi, dans [Royer *et al.*, 2004], les mouvements sont déduits de l'erreur de pose mesurée entre l'image courante et la prochaine image de la séquence à rejouer. Dans [Rasmussen and Hager, 1996, Burschka and Hager, 2001], le déplacement est assuré en imposant aux primitives de suivre les trajectoires images observées lors de la phase d'apprentissage.

De nombreux travaux s'appuient sur la connaissance de la topologie de l'environnement pour définir un chemin dans celui-ci, sans pour autant rejouer une trajectoire préalablement apprise. Ces approches dites de planification se différencient encore une fois par la représentation interne de l'environnement (diagrammes de Voronoï, cartes probabilistes ou décomposition en cellules libres) [Latombe, 1993, Laumond, 1998]. Ces approches permettent de considérer des systèmes robotiques complexes, possédant de nombreux degrés de liberté. Cependant, ces techniques se limitent généralement à la génération de la trajectoire à réaliser sans détailler le problème de réalisation de cette trajectoire par le robot, où peuvent se poser des problèmes de calcul de pose, ou encore de variation de l'environnement depuis la planification. Le lecteur intéressé pourra se référer aux ouvrages de références cités ci-dessus pour obtenir plus de détails sur ces techniques.

1.2. Approche basée apparence

Les approches basées apparence, ou topologiques, présentent l'avantage de ne pas imposer la connaissance du modèle 3D de la scène, et proposent de travailler directement dans l'espace capteur. L'ensemble des descriptions discrètes obtenues sont organisées sous la forme d'un graphe de connectivité. Les noeuds représentent les signatures capteurs obtenues depuis les différentes positions d'observation de l'environnement. Un arc relie deux noeuds si le robot a la possibilité de se mouvoir de manière autonome entre les deux positions sous-jacentes, en utilisant les données extraites de ces deux descriptions locales.

Dans les applications basées vision, les descriptions locales sont les images acquises lors de la phase d'apprentissage. Dans ce cadre, la localisation initiale du robot est généralement traitée comme un problème de recherche d'images, et s'appuie donc sur les techniques développées dans ce domaine de recherche très actif [Smeulders *et al.*, 2000]. De manière générale, un score de similarité est calculé entre la vue requête (la vue à localiser) et les différentes images de la base. Les descripteurs utilisés sont nombreux. Certains sont globaux, comme les histogrammes de couleurs [Zhou *et al.*, 2003], voire même le gradient de l'image [De La Torre and Black, 2001, Košeck *et al.*, 2003]) ou encore l'image brute, comme dans les travaux de [Jones *et al.*, 1997, Matsumoto *et al.*, 2000]. L'utilisation de descripteurs locaux permet, en associant à une image plusieurs descripteurs, de rendre la localisation plus robuste aux potentielles perturbations locales de l'image (causées par exemple par l'apparition ou la disparition d'un objet de la scène). Citons

comme exemple de descripteurs locaux la caractérisation SIFT (pour *Scale Invariant Feature Transform*) de points de Harris [Lowe, 2004], ou encore les invariants photométriques [Remazeilles *et al.*, 2004].

Différentes stratégies sont là-encore possibles pour réaliser la navigation. Une des approches consiste à enregistrer un mouvement spécifique pour chaque image de la base [Jones *et al.*, 1997, Matsumoto *et al.*, 2000]. À chaque pas de temps, le robot réalise le mouvement associé à la vue de la base dont il est le plus proche. L'un des problèmes de cette approche est qu'elle ne permet pas de prendre en compte une potentielle déviation du chemin préalablement appris, ce qui peut alors rendre les mouvements enregistrés inefficaces. Dans [Argyros *et al.*, 2001, Blanc *et al.*, 2005], la navigation est assurée en convergeant successivement par asservissement visuel vers les différentes vues intermédiaires. Cependant, ce type d'approche impose d'avoir une représentation de l'environnement suffisamment précise pour pouvoir fournir une trajectoire satisfaisante quelles que soient les images initiales et désirées considérées. De plus, il est en général inutile de converger vers ces images intermédiaires surtout si ces convergences locales ne sont pas nécessaires pour atteindre la position désirée.

Notons pour finir que certaines approches s'appuient sur une représentation mixte de l'environnement, comme dans [Hayet *et al.*, 2002] où un diagramme de Voronoï construit par un laser est annoté par des images de posters observés par une caméra embarqué au sein de l'environnement intérieur. Ces approches intermédiaires permettent de combiner les avantages de chacune des approches : la vision est ainsi utilisée pour localiser le robot lors de la navigation dans de longs couloirs, où les informations issues du laser s'avèrent peu précises. Si la combinaison de différents capteurs et d'informations sensorielles apparaît comme une solution d'intérêt, elle rend cependant la phase de description de l'environnement plus complexe.

1.3. Approche proposée : une navigation topologique qualitative

Les travaux présentés dans cet article appartiennent à la deuxième famille d'approche. Ainsi, nous nous affranchissons d'une phase initiale de reconstruction 3D, et nous n'imposons

pas d'estimer la pose absolue de la caméra durant la navigation. Les Figures 1 et 2 présentent schématiquement les opérations permettant au système de se définir un chemin topologique pour atteindre une position donnée. La première figure illustre la localisation initiale qui est réalisée avant le début de la tâche proprement dite de navigation. Dans ce cadre, aucune hypothèse *a priori* n'est effectuée quant à la pose du système robotique (on notera d'ailleurs que de nombreux travaux en font de même [Košeck *et al.*, 2003, Zhou *et al.*, 2003]). La seule information utilisée est une base d'images acquise durant une phase d'apprentissage (base entourée en orange sur la figure). La localisation revient alors à déterminer les vues de la base qui sont, en terme de contenu, les plus similaires à l'image dite requête (image initiale ou désirée). Ces images sont pointées par des flèches bleues sur la figure.

Une fois que les images initiale et désirée ont été mises en relation avec certaines des vues de la base, l'étape suivante consiste à restreindre la base d'images à un ensemble de vues décrivant l'environnement que doit traverser le système robotique durant sa navigation. La Figure 2 illustre ce traitement. La structuration de cette base sous forme de graphe permet de résoudre facilement cette recherche. Ainsi, à chaque noeud du graphe est associée une image de la base, et un arc entre deux noeuds caractérise un déplacement entre deux vues que pourrait réaliser automatiquement le système robotique. La localisation initiale du robot a permis de mettre en relation les images initiale et désirée avec certaines vues de la base. Il est donc alors possible d'incorporer ces deux vues au sein du graphe. La restriction de la base d'images aux vues caractérisant l'environnement que doit parcourir le système n'est alors rien d'autre qu'une recherche de chemin dans le graphe (ce chemin est illustré en rouge sur la Figure 2).

Dans le schéma de navigation proposé, le système robotique n'est pas imposé de converger vers les différentes vues du chemin, ce qui lui donne ainsi plus de flexibilité (et donc de réactivité si nécessaire) durant la navigation. Le contrôle des mouvements du système est réalisé par asservissement visuel. Nous proposons de plus une nouvelle approche d'asservissement qualitatif dans lequel les informations visuelles utilisées pour contrôler le système sont régulées pour atteindre non plus une valeur particulière, mais un intervalle de confiance.

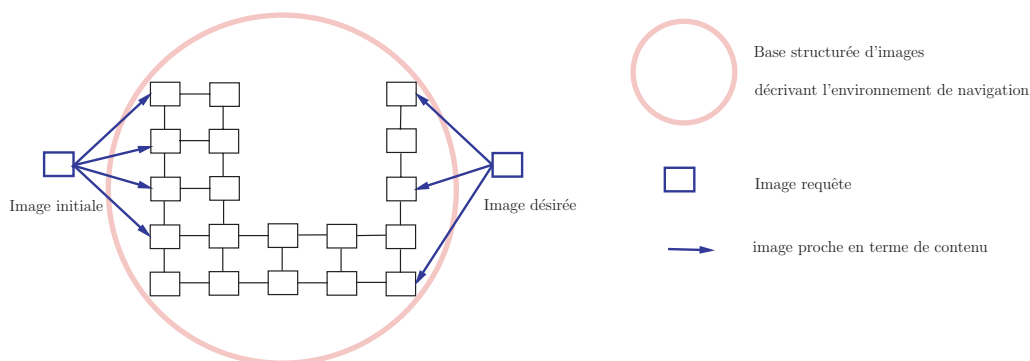


Figure 1. Localisation qualitative par recherche d'images.

La section suivante présente succinctement les phases de localisation et de recherche de chemin dans la base. La Section 3 s'intéresse à la navigation proprement dite. La Section 4 présente des résultats expérimentaux obtenus en simulation et avec un bras articulé. Enfin, la dernière section conclut cet article.

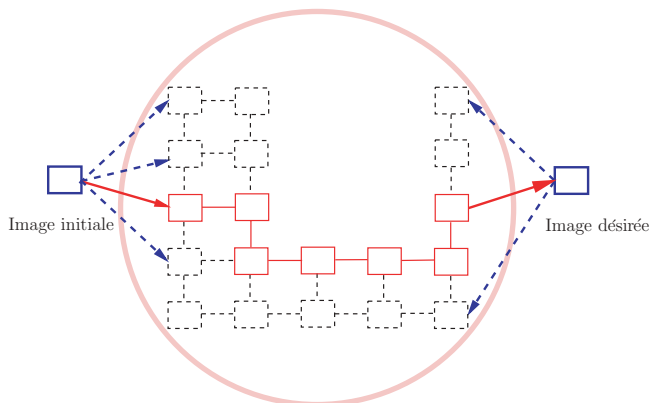


Figure 2. Recherche d'un chemin d'images. Le robot utilisera ce chemin pour contrôler ses mouvements.

2. Localisation du robot et détermination du chemin

2.1. Localisation qualitative

L'emploi de techniques issues du domaine de la recherche d'images est très répandu dans le cadre de la localisation robotique basée vision. Classiquement, la recherche d'images est réalisée en détectant et en caractérisant des amers visuels dans les images de la base. La reconnaissance d'une image donnée revient alors à effectuer cette même caractérisation sur cette vue, et à mettre en relation les descripteurs obtenus avec ceux de la base, au moyen d'une métrique de similarité. Le niveau de

similarité obtenu par chacune des images de la base permet ensuite de définir quelles sont les vues qui sont les plus proches de l'image requête.

La méthode employée ici s'appuie sur une caractérisation locale des images. L'avantage vis-à-vis des caractérisations dites globales est que la reconnaissance est plus robuste aux perturbations locales que peut subir l'information capteur (comme par exemple l'apparition ou la disparition d'un des objets de la scène). Cette propriété est très intéressante dans le sens où l'environnement a très bien pu évoluer localement depuis la phase d'apprentissage.

L'algorithme utilisé a été développé par Berrani [Berrani *et al.*, 2003] et est basé sur des travaux de Schmid [Schmid and Mohr, 1997]. Les descriptions locales sont des invariants photométriques décrivant le signal dans le voisinage d'un point d'intérêt (point caractérisé par une forte variation bidimensionnelle du signal [Harris and Stephens, 1988, Shi and Tomasi, 1994]). Pour chaque descripteur de l'image requête, les descripteurs de la base qui sont les plus similaires (suivant une métrique choisie) donnent une voix à leurs images d'appartenance. En additionnant ces résultats locaux, on obtient les images qui possèdent le plus de descripteurs communs avec l'image à reconnaître. Les expériences présentées ci-dessous montrent que ce principe de recherche d'images est aussi très efficace pour réaliser une localisation basée vision de systèmes robotiques.

Pour illustrer cette approche, nous utilisons une base d'images constituée de deux séquences acquises par un système stéréoscopique embarqué dans un véhicule. Pour chacune de ces séquences, une image sur 10 est conservée, ce qui correspond à une base de 280 vues. La Figure 3 présente un échantillon des prises de vue constituant cette base. Dans les deux expériences présentées sur les Figures 4 et 5, l'image recherchée est une des images acquises lors de l'apprentissage, mais n'est pas présente dans la base de l'environnement. Les images retrouvées sont présentées avec leurs scores obtenus lors de la recherche des k -plus proches voisins.

Sur la première expérience, nous pouvons observer que les trois premières images présentent un score nettement plus important que les suivantes. Les images les plus proches de la requête qui sont stockées dans la base sont les vues (b) et (c) (elles ont été



Figure 3. Illustration de la variété des images de la base.



Figure 4. Localisation qualitative par recherche d'images. (a) : image requête, (b) à (i) : plus proches vues trouvées dans la base. Les nombres entre parenthèses représentent le nombre de voix obtenues par chacune des vues. Les images sont de dimension 728×440 .

acquises respectivement avant et après l'image requête). Notons que des vues qui sont visuellement complètement différentes de l'image requête (image (e) par exemple) sont mieux classées que d'autres qui nous semblent pourtant plus pertinentes, comme la vue (g). Une explication de ce phénomène est donnée dans l'étude de la deuxième expérience qui suit.

Sur les images présentées ici, certains éléments ne caractérisent pas l'environnement de navigation. La plupart sont mobiles, comme les piétons ou bien les voitures. Pour montrer la robustesse du schéma de localisation, l'expérience suivante utilise seulement les points de l'image requête qui caractérisent la scène statique (dessinés en rouge sur la Figure 5(a)). Tous les points extraits des images de la base sont utilisés, y compris ceux associés aux objets mobiles. On peut considérer ces derniers comme du bruit venant perturber le processus de localisation. La Figure 5 montre que les images les plus proches sont toujours détectées (les vues b et e sont les images de la base prises juste avant et juste après l'image considérée comme requête).

Notons que parmi les 246 points initiaux de l'image requête (en rouge et vert), seulement 147 ont été employés pour la localisation. De plus, sur les 100 points de plus haute courbure (c'est-à-dire les points les plus discriminants), 63 étaient associés à des objets mobiles. La localisation reste donc performante même si le nombre de points utilisé est moins important. Le fait d'utiliser des descripteurs locaux, moins sensibles aux perturbations locales de l'image, permet d'obtenir de tels résultats.

Sur la figure 5, on note cependant que la vue (c), qui décrit un environnement complètement différent de celui de la vue (a), se

trouve mieux classée que les vues (d) et (e) qui semblent pourtant visuellement plus pertinentes. Une première explication réside dans la simplicité de la mise en relation des descripteurs, à savoir la recherche des k plus proches vecteurs. Celle-ci est purement locale, et ne prend pas en compte des contraintes de mouvement rigide. Rien ne permet ainsi d'assurer qu'une association réalisée est valide ou non (de nombreuses associations entre les vues (a) et (e) doivent ainsi être erronées). La sommation du nombre des associations réalisées vers chacune des images de la base permet cependant de relever les vues qui semblent partager le plus de primitives en commun avec la vue requête. Mais encore une fois, aucun facteur ne permet à ce niveau d'assurer que les images classées possèdent une réelle relation avec la vue d'intérêt. Dans cette expérience, le retrait de nombreux points de forte saillance, c'est-à-dire de points très discriminants, peut aussi expliquer le classement de (b). De plus, on observe que les scores obtenus par les images sont beaucoup plus faibles dans la deuxième expérience. La plus petite quantité de points décrivant l'image requête ne permet ainsi pas aux images semblables de se distinguer des autres vues de la base. En effet, il n'y a pas dans cette deuxième expérience une séparation notable permettant de dégager des vues plus fortement sollicitées (comme c'est le cas dans l'expérience 1 avec les vues (b), (c) et (d)).

Ces cas de figure sont détectés et traités lors de l'ajout de la vue dans le graphe décrivant l'environnement. Comme le décrira la phase suivante, cet ajout impose la connaissance des primitives en commun entre la vue requête et ses images proches, et ce de manière robuste. L'utilisation d'un algorithme d'appariement vérifiant des contraintes globales (via la géométrie épipolaire),



Figure 5. Localisation qualitative par recherche d'images. (a) : image requête. Les points de Harris sont tracés. Les points verts correspondent à des objets mobiles de la scène, et ne sont pas utilisés lors de la localisation. Images (b) à (i) : plus proches vues trouvées dans la base d'images. Le plus faible nombre de points de l'image requête utilisé, ainsi que leurs saillances plus faibles, expliquent les scores très proches obtenus.

permet de rejeter les vues supposées proches par la phase d'indexation mais qui ne sont pas cohérentes d'un point de vue des contraintes rigides que doivent respecter deux prises de vue d'une même scène.

Pour finir, la localisation réalisée peut être qualifiée de qualitative, puisque la position 3D du système robotique n'est pas estimée. Nous verrons dans la suite que cette localisation qualitative est suffisante pour réaliser avec succès la phase de navigation autonome.

2.2. Recherche du chemin d'images

Une fois que l'image initiale est reliée à la base d'images, le système robotique doit délimiter l'environnement qu'il doit traverser afin d'atteindre sa position désirée (voir la Figure 2). Plus exactement, il doit définir les primitives qui devraient être observées par la caméra durant la navigation. Ces primitives seront les seules informations employées non seulement pour localiser le robot, mais aussi pour contrôler ses mouvements.

La représentation topologique de l'environnement permet de traiter ce problème comme une recherche de chemin dans le graphe associé à la base. En général, les graphes utilisés sont non valués, ce qui signifie que tous les mouvements associés à un arc entre deux noeuds de la base sont considérés de difficulté équivalente. En effet le graphe correspond généralement à un ensemble de séquences que le système est censé rejouer durant

la navigation [Matsumoto *et al.*, 2000, Blanc *et al.*, 2005]. Cependant, il peut être pratique de proposer au système différents chemins pour parvenir à une position donnée, afin de lui proposer plusieurs alternatives. Le système robotique doit alors être capable de choisir le chemin le plus aisé à réaliser. Une solution naïve consisterait à utiliser directement le nombre de primitives appariées entre deux vues pour caractériser la facilité d'un déplacement [Remazeilles *et al.*, 2004]. En effet, plus le nombre de correspondances entre deux images est grand, plus le mouvement entre ces deux vues est petit, ce qui peut en effet être considéré comme un critère de facilité. Néanmoins, suivant le type de système robotique employé, certains mouvements peuvent être coûteux même si le mouvement est petit. Par exemple, un mouvement latéral est très coûteux pour une voiture, puisqu'il impose de réaliser des manœuvres. Il est donc nécessaire de prendre en compte la difficulté du mouvement considéré dans la valuation.

Nous proposons une fonction adaptée au cas des véhicules mobiles comme une voiture, où la caméra embarquée observe dans la direction du mouvement, comme illustré sur la Figure 6. Cette fonction de coût repose sur le mouvement partiel (\mathbf{R}, \mathbf{b}) reliant les poses associées aux deux vues, avec \mathbf{R} la matrice de rotation, et $\mathbf{b} = (b_x \ b_y \ b_z)$ le vecteur de translation à un facteur d'échelle près ($\mathbf{b} = \alpha \mathbf{t}$). Ce mouvement partiel peut être estimé à partir des correspondance de points (voir la Section 3.2 pour plus de détails).

Le coût d'un arc $\gamma(x_i, x_j)$ entre deux noeuds x_i et x_j du graphe est ainsi défini comme suit :

$$\gamma(x_i, x_j) = \begin{cases} \infty & \text{si } n_{i,j} < 8, \\ \gamma_\infty & \text{si } b_z = 0, \\ 1 + \frac{|b_x| + \kappa\theta|u_y|}{|b_z|} & \text{sinon,} \end{cases} \quad (1)$$

où :

- ∞ est le coût maximal entre deux noeuds non reliés par un arc (le nombre de points appariés entre les images x_i et y_i , noté $n_{i,j}$ doit être au moins égal à huit pour effectuer l'estimation du mouvement partiel).

- γ_∞ est une constante maximale associée aux mouvements purement latéraux (qui arrivent très rarement en pratique).

- $\theta \mathbf{u} = \theta (u_x \ u_y \ u_z)$ représente la rotation entre les deux vues (\mathbf{u} est l'axe de rotation, et θ l'angle de rotation autour de cet axe).

- κ est un scalaire positif pondérant l'importance des rotations vis-à-vis des translations.

Cette fonction pénalise les mouvements latéraux (le long de l'axe \vec{x}) et les trop grandes rotations (autour de l'axe \vec{y}). Elle favorise les mouvements le long de l'axe optique.

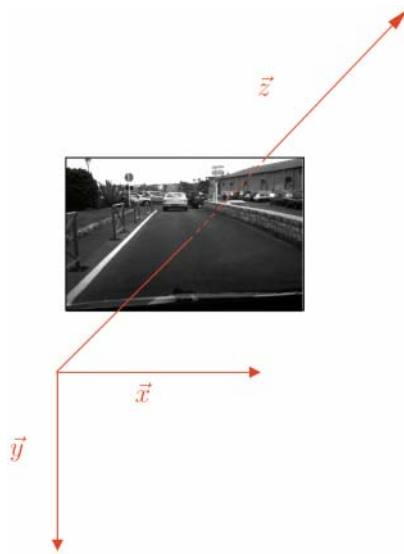


Figure 6. Orientation de la caméra embarquée.

La Figure 7 présente les résultats d'une recherche de chemin sur une base associée à une séquence acquise par un système stéréoscopique embarquée dans une voiture se déplaçant sur une route. Chaque point bleu correspond à la pose d'une image de la base. Les arcs du graphe ne sont pas présentés sur cette figure. Notons que des arcs relient les images proches sur les deux parcours (sommairement, une image est connectée vers les trois images suivantes du même parcours, et avec les trois images acquises par l'autre caméra du système stéréoscopique). En rouge sont tracées les images de la séquence qui sont sélectionnées par l'algorithme de recherche de chemin entre deux noeuds du graphe. La figure de gauche illustre tout le chemin, alors que celle de droite réalise un gros plan sur le début du che-

min. Sur la figure de droite, on peut noter que certains noeuds entre ceux sélectionnés ne sont pas compris dans le chemin final. En effet, si l'algorithme de recherche de chemin détermine un chemin entre deux noeuds, il choisit celui permettant de minimiser le coût global de celui-ci. La sélection de tous les noeuds entre les positions initiale et désirée aurait amené à un coût plus important du chemin.

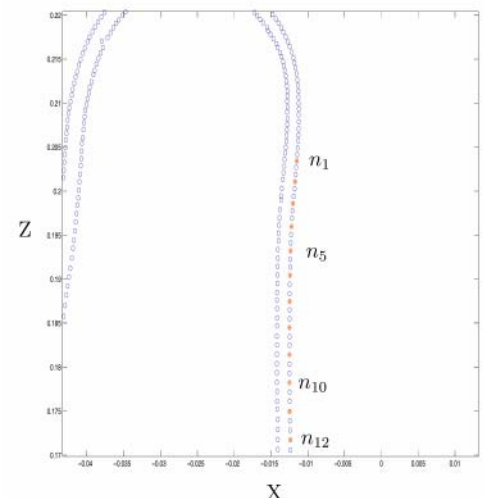
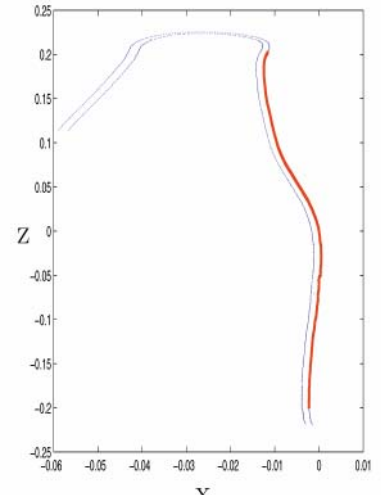


Figure 7. Illustration de la recherche de chemin. Les points bleus correspondent aux images de la base (la position 3D des vues a été fournie par *Éric Royer [Royer et al., 2004]*). La ligne rouge relie les images du chemin sur la figure. L'image de droite se focalise sur le début du chemin. Les images sélectionnées y sont représentées par un rond rouge.

Cette section a donné quelques clés concernant la détermination du chemin d'images. La principale caractéristique de ce chemin est qu'il définit de manière continue l'environnement que doit observer la caméra durant la navigation. La section suivante présente comment sont déduits les mouvements du robot à partir de ce chemin et de l'image fournie par la caméra.

3. Navigation par asservissement visuel qualitatif

Dans la suite, le chemin d'images est noté ψ_0, \dots, ψ_N . ψ_0 est l'image acquise par la caméra avant le début des mouvements (image initiale). ψ_N est la vue que doit fournir la caméra une fois le déplacement réalisé (image désirée). \mathcal{M}_i désigne l'ensemble de points $({}^i\mathbf{x}_j, {}^{i+1}\mathbf{x}_j)$ qui ont été appariés entre les vues ψ_i et ψ_{i+1} du chemin.

3.1. Boucle de contrôle

Chaque ensemble \mathcal{M}_i correspond à un ensemble de points visibles entre deux images de la base. Ces primitives décrivent donc l'environnement situé entre les deux positions associées de la caméra. Afin de rejoindre la position désirée, le système robotique va donc devoir successivement observer les différents emplacements décrits par les ensembles \mathcal{M}_i . Une tâche de navigation peut donc être formulée comme suit :

Soit \mathcal{M}_i l'ensemble de points appariés entre les images ψ_i et ψ_{i+1} du chemin. Supposons que cet ensemble est partiellement ou totalement visible par la caméra depuis sa position courante. Étant donné un ensemble de fonctions de coût décrivant la projection de cet ensemble sur le plan image courant ψ_t , les mouvements du système robotique ont pour but de faire converger ces mesures visuelles vers des intervalles de confiance, afin de rapprocher le robot de positions où il peut observer les points de l'ensemble suivant \mathcal{M}_{i+1} .

Il est important de noter que déplacer le robot afin d'observer les prochains ensembles d'amers visuels n'impose pas de converger vers les différentes images du chemin. Dans ce formalisme, la loi de commande est définie de sorte que les mouvements du système permettent de le déplacer vers des zones de l'espace où les conditions d'observation des primitives sont considérées satisfaisantes.

La Figure 8 présente la boucle générale de commande utilisée pour définir les mouvements du robot. Suite à l'acquisition d'une nouvelle image ψ_t par la caméra, les différentes étapes de traitement sont :

1. **Suivi de points** : les points ${}^{t-1}\mathbf{x}_j$ visibles dans l'image précédente ψ_{t-1} sont suivis dans la nouvelle image ψ_t afin de mettre à jour leur position.
2. **Transfert de points** : les points initialement détectés sur le chemin d'images sont localisés sur le plan image courant. L'estimation de leur position est effectuée par reprojection depuis le chemin d'images, en utilisant des techniques issues de la géométrie épipolaire. Ce transfert permet de détecter si de nouvelles primitives sont rentrées dans le champ de vision de la caméra suite au mouvement du système robotique.

3. **Mise à jour des points visibles** : l'ensemble des points visibles ${}^t\mathbf{x}_j$ est mis à jour en rajoutant les points visibles localisés par l'étape précédente de transfert. Les points rajoutés doivent se projeter dans le cadre image. De plus, une forte variation bidirectionnelle du signal, similaire à celle définissant ce point dans la vue du chemin où il été défini, doit être observée.

4. **Sélection de l'ensemble d'intérêt** : parmi tous les ensembles \mathcal{M}_i dont certaines primitives sont déjà visibles, le plus avancé sur le chemin est choisi. Il constitue l'ensemble de primitives que la caméra doit chercher à observer dorénavant.

5. **Mise à jour de la loi de commande** : les mouvements sont définis en fonction de l'ensemble d'intérêt choisi. Ces mouvements permettent de déplacer le système vers une zone de l'environnement où les conditions d'observation de cet ensemble sont jugées meilleures.

L'étape de suivi (1) peut être réalisé dans une application réelle par un algorithme de suivi différentiel, comme [Jin *et al.*, 2001]. Une méthodologie pour réaliser le transfert de points est décrite ci-dessous. L'étape 5 fait le sujet de la section 3.3.

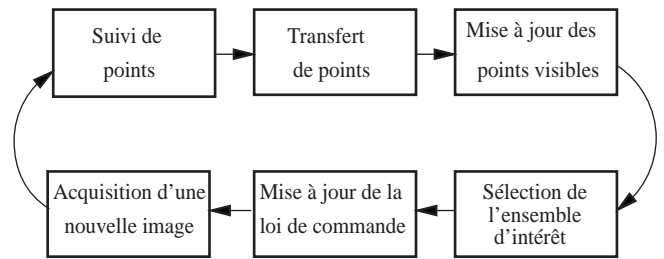


Figure 8. Boucle de contrôle employée.

3.2. Relation géométrique entre les prises de vue

Soient ${}^1\mathbf{x}_p$ et ${}^2\mathbf{x}_p$ les projections d'une même primitive 3D dans deux images ψ_1 et ψ_2 . Ces coordonnées peuvent être mises en relation par la matrice d'homographie ${}^2\mathbf{H}_{p_1}$ [Hartley and Zisserman, 2000] :

$${}^2\mathbf{x}_p \propto {}^2\mathbf{H}_{p_1} {}^1\mathbf{x}_p + \beta_{1,j} \mathbf{e}_2, \quad (2)$$

avec :

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

$$\text{et } \beta_{1,j} = \frac{d_j}{Z_1 d_\pi} \quad (3)$$

\mathbf{K} contient les paramètres intrinsèques de la caméra, et \mathbf{e}_2 désigne l'épipôle de la deuxième caméra. $({}^2\mathbf{R}_1, {}^2\mathbf{t}_1)$ est le mouvement rigide entre les deux poses. Cette rotation et cette translation (à un facteur d'échelle près) peuvent être extraites de l'homographie [Faugeras and Lustman, 1988]. L'homographie est relative à un plan de référence π . \mathbf{n} désigne sa normale, et d_j la distance signée entre le point 3D et ce plan (voir la Figure 9). Si tous les points appartiennent à ce plan de référence, seules

quatre correspondances sont nécessaires pour estimer la matrice d'homographie, et $\beta_{i,j} = 0$. Dans le cas général, huit appariements sont requis [Malis and Chaumette, 2000, Hartley and Zisserman, 2000].

La parallaxe $\beta_{1,j}$ est déduite de l'équation précédente :

$$\beta_{1,j} = -\frac{({}^2\mathbf{H}_{p_1} {}^1\mathbf{x}_p \wedge {}^2\mathbf{x}_p)^\top ({}^2\mathbf{c}_2 \wedge {}^2\mathbf{x}_p)}{\|{}^2\mathbf{c}_2 \wedge {}^2\mathbf{x}_p\|^2} \quad (4)$$

On peut noter dans l'équation (3) que la parallaxe est indépendante de la position de la seconde caméra. Cependant, puisque l'épipole n'est connue qu'à un facteur d'échelle près, l'équation (2) déduite des données images est plutôt de la forme :

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

où $\beta_{\alpha 1_j} = \alpha \beta_{1_j}$. La parallaxe dépend donc du facteur d'échelle de la translation. Pour faire disparaître le scalaire α , [Shashua and Navab, 1996] propose de *normaliser* l'homographie par rapport à un point de référence \mathcal{X}_0 choisi tel que $\mathcal{X}_0 \notin \pi$:

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

ce qui permet d'obtenir une parallaxe indépendante du facteur d'échelle :

$$\beta'_{1_j} = \frac{\beta_{\alpha 1_j}}{\beta_{\alpha 1_0}} = \frac{\alpha d_j}{Z_j d_\pi} \frac{Z_0 d_\pi}{\alpha d_0} = \frac{d_j Z_0}{d_0 Z_j}$$

Ainsi, si l'on connaît l'homographie ${}^3\mathbf{H}_{p_1}$, définie depuis la même image de référence ψ_1 et projetant les points vers une troisième vue ψ_3 , et si cette homographie est aussi mise à l'échelle par rapport au même point \mathcal{X}_0 , il devient alors possible de prédire la position dans ψ_3 de n'importe quel point apparié entre les vues ψ_1 et ψ_2 :

$${}^3\mathbf{x}_{p_j} \propto {}^3\mathbf{H}'_{p_1} {}^1\mathbf{x}_{p_j} + \beta'_{1_j} {}^3\mathbf{c}_3, \quad (5)$$

Ce principe peut être utilisé pour réaliser le *transfert d'images* entre les différentes images du chemin et la vue courante.

Notons de plus que l'homographie permet de déterminer quelques grandeurs caractérisant la scène observée, comme le ratio entre les profondeurs Z_1 et Z_2 d'un point 3D [Malis and Chaumette, 2000] :

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

et le ratio entre la profondeur Z_2 et la distance d_1 :

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

avec ${}^2\mathbf{t}_1 / Z_1 = \tau {}^2\mathbf{x}_n - {}^2\mathbf{R}_1 {}^1\mathbf{x}_n$. Ces relations seront utiles par la suite.

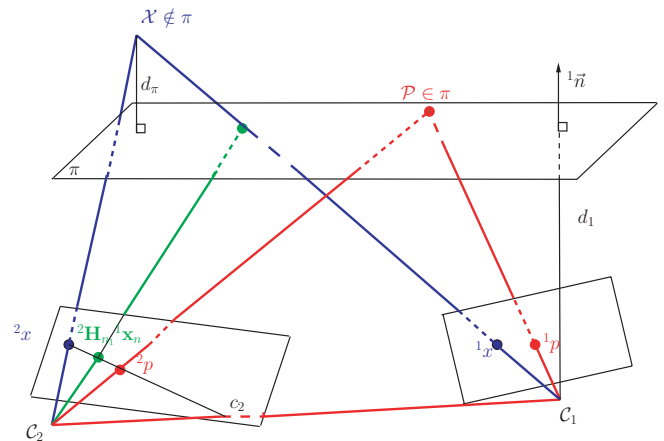


Figure 9. Géométrie relative à la matrice d'homographie.

3.3. Calcul de la loi de commande

3.3.1. Asservissement visuel qualitatif

La nouvelle loi de commande présentée dans cet article peut être considérée comme un *asservissement visuel qualitatif*, en comparaison avec le schéma classique de l'asservissement visuel. Dans ce dernier, le but est de minimiser une erreur mesurée entre la position courante d'un ensemble de primitives visuelles \mathbf{s} et leurs positions désirées \mathbf{s}^* . Puisque \mathbf{s} dépend de la pose de la caméra, la pose désirée \mathbf{p}^* est atteinte lorsque l'erreur mesurée est nulle, soit $\mathbf{s} = \mathbf{s}^*$. Pour ce faire une loi de commande classique est donnée par [Espiau *et al.*, 1992] :

$$\mathbf{v} = -\lambda \mathbf{L}_s^+ (\mathbf{s} - \mathbf{s}^*), \quad (7)$$

où \mathbf{v} est la vitesse de la caméra envoyée au contrôleur bas-niveau du robot, λ est un gain permettant de régler la vitesse de convergence du système, et \mathbf{L}_s^+ la pseudo-inverse de la matrice d'interaction associée à \mathbf{s} et définie telle que $\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}$.

Dans l'approche proposée, aucune mesure désirée ne peut être spécifiée, puisque le robot n'est pas imposé de converger vers les différentes images intermédiaires du chemin. Le robot est simplement contraint de se déplacer dans des zones où la projection des points de l'ensemble \mathcal{M}_i est considérée comme satisfaisante.

Ainsi, le robot est seulement contraint d'atteindre une zone où $\mathbf{s} \in [\mathbf{s}_{min}; \mathbf{s}_{max}]$. Ceci est réalisé en définissant des fonctions de coût \mathcal{V} adéquates, telles que leurs minima correspondent à des positions où les primitives visuelles appartiennent à $[\mathbf{s}_{min}; \mathbf{s}_{max}]$. Dans ce contexte, le gradient de ces fonctions $\nabla \mathcal{V}(\mathbf{p})$ se substitue à \mathbf{s} dans l'équation (7). La mesure désirée \mathbf{s}^* devient alors $\nabla \mathcal{V}(\mathbf{p})^*$, grandeur définie comme nulle. La loi de commande utilisée à la place de (7) est ainsi :

$$\mathbf{v} = -\lambda \mathbf{L}_{\nabla \mathcal{V}}^+ \nabla \mathcal{V}, \quad (8)$$

où $\mathbf{L}_{\nabla \mathcal{V}}$ est la matrice d'interaction associée au gradient de \mathcal{V} .

Les sections 3.3.2, 3.3.3 et 3.3.4 détaillent les différentes fonctions $\mathcal{V}(\mathbf{p})$ proposées, ainsi que les grandeurs $\nabla\mathcal{V}(\mathbf{p})$ et $\mathbf{L}_{\nabla\mathcal{V}}$ qui sont utilisées pour calculer la loi de commande. La section 3.3.5 décrit cette mise à jour.

3.3.2. Progression le long du chemin

La première fonction de coût permet de déplacer le système robotique vers l'avant. Lorsque l'on considère une caméra perspective, comme illustré sur la figure 10(a), la position de la projection d'un point 3D $\mathcal{X} = (X, Y, Z)$ est inversement proportionnelle à sa profondeur Z , étant donné que $\mathbf{x} = (X/Z, Y/Z)$. Ainsi, plus la profondeur du point est grande, plus sa projection est proche du centre de projection de la caméra. Le même raisonnement tient si l'on déplace la caméra le long de son axe optique. En effet, si l'on considère un mouvement de translation t_z entre deux positions de caméra, alors la projection d'un point devient dans la seconde vue $(X/(t_z + Z), Y/(t_z + Z))$. Ainsi, plus la translation est grande, plus le point se rapproche du centre de projection. Ce fait est illustré sur la Figure 10(b), où ψ_0 peut être associée à la vue courante fournie par la caméra, et ψ_1 à la prochaine vue du chemin. On voit facilement que la zone observée par la caméra ψ_1 occupe une superficie beaucoup plus petite dans la prise de vue courante. C'est cette information qui va être employée pour contrôler les mouvements le long de l'axe optique.

Pour décrire la surface de projection des primitives, nous utilisons une mesure basée sur les moments centrés d'ordre deux :

$$a = \mu_{02} + \mu_{20},$$

avec, pour un ensemble de n points :

$$\mu_{ij} = \sum_{k=0}^n (x_k - x_g)^i (y_k - y_g)^j,$$

où (x_g, y_g) est le centre de gravité image de ces points. Plus les points 3D se rapprochent du plan image de la caméra, plus cette mesure grandit. Intuitivement, a est fortement relié à la surface de projection de ces points sur le plan image.

La mesure suivante compare la valeur courante de a avec celle mesurée sur la prochaine image du chemin a^* :

$$a_n = \sqrt{\frac{a^*}{a}} \tag{9}$$

Puisque le robot n'est pas contraint à passer explicitement par chaque image du chemin, nous ne cherchons pas à obtenir exactement la même mesure a^* , mais plutôt une mesure de même ordre de grandeur. Soit $p \in [0, 1]$ le pourcentage de liberté autorisée autour de a_n^* . Une mesure satisfaisante est telle que :

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

L'appartenance à cet intervalle peut être décrite par la fonction suivante :

$$\mathcal{V}_{a_n}(a_n) = \begin{cases} \frac{1}{2} (a_n - a_M)^2 & \text{si } a_n > a_M \\ \frac{1}{2} (a_m - a_n)^2 & \text{si } a_n < a_m \\ 0 & \text{sinon} \end{cases} \tag{10}$$

Afin d'obtenir une transition continue entre les trois cas, nous utilisons la fonction suivante :

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

où (voir la Figure 11) :

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

$h_k(x)$ est la fonction arc-tangente normalisée sur $[0; 1]$. Elle définit un palier entre les valeurs 0 et 1. La transition se réalise

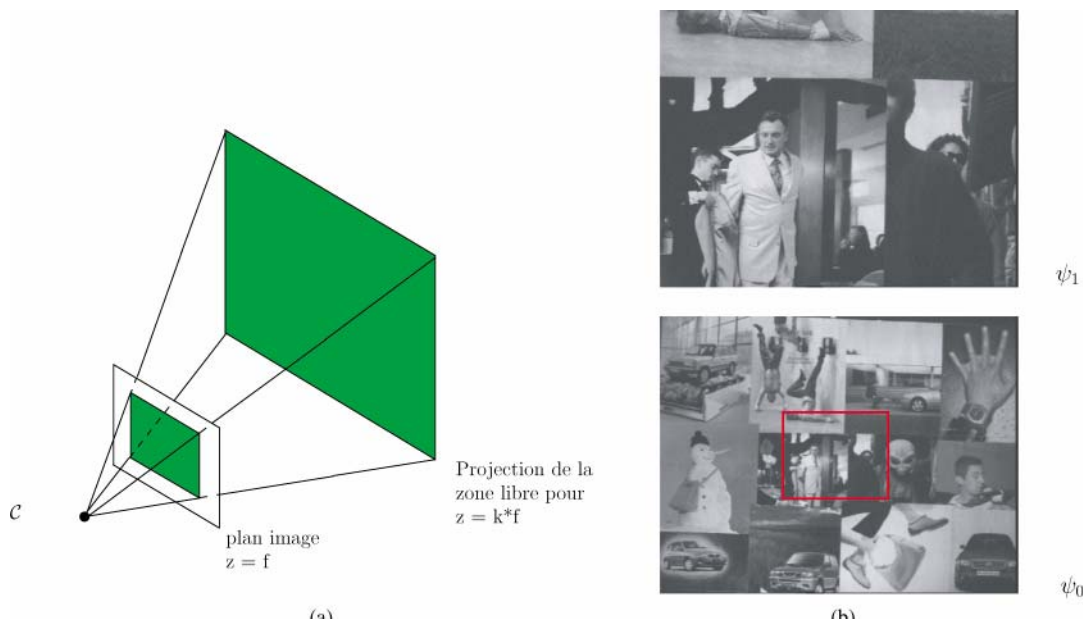


Figure 10. Mouvements le long de l'axe optique : (a) cône de visibilité de la caméra, (b) comparaison de la zone couverte par le champ de vision d'une caméra suite à un déplacement le long de l'axe optique.

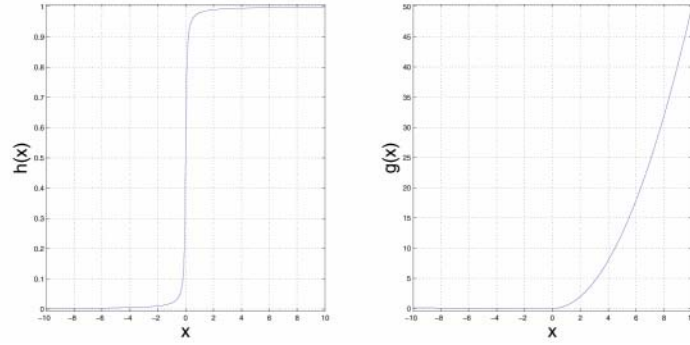
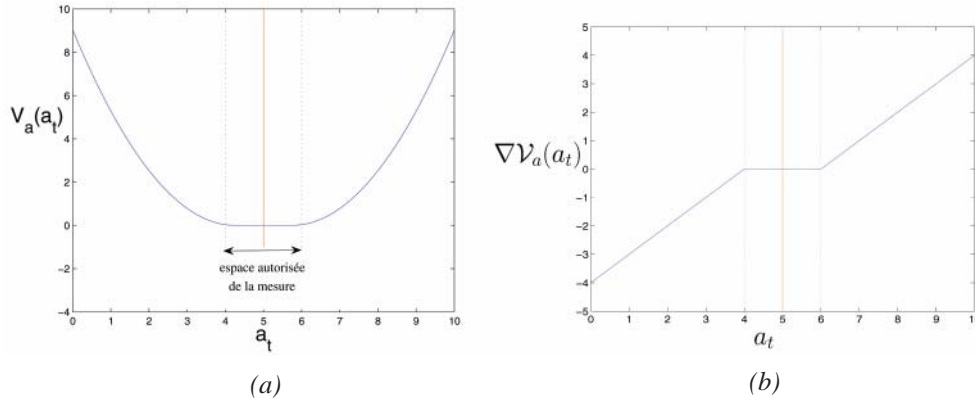

 Figure 11. Fonctions h et g utilisées pour lisser la fonction de coût (voir l'équation (12)).


Figure 12. Contrôle des mouvements le long de l'axe optique : (a) fonction utilisée, (b) son gradient.

pour $x = 0$. Le scalaire k permet de régler la courbure de cette transition. Comme on peut le voir sur la Figure 12, la fonction \mathcal{V}_{a_n} ainsi définie est nulle lorsque la mesure a_n appartient à l'intervalle de confiance, et s'accroît d'autant plus que a_n s'éloigne de cet intervalle.

L'erreur associée à \mathcal{V}_{a_n} est déterminée comme suit :

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

où $\nabla_{a_n} \mathcal{V}_{a_n}$ est :

$$\begin{aligned} \nabla_{a_n} \mathcal{V}_{a_n} &= (a_n - a_M) h(a_n - a_M) + \mathcal{O}(a_n - a_M) \\ &\quad + (a_n - a_m) h(a_m - a_n) - \mathcal{O}(a_m - a_n), \end{aligned}$$

avec :

$$\mathcal{O}(x) = \frac{kx^2}{2(1 + k^2\pi^2x^2)} \quad (14)$$

Si l'on s'affranchit des fonctions h et \mathcal{O} , une bonne approximation du gradient est :

$$\nabla_{a_n} \mathcal{V}_{a_n} = \begin{cases} a_n - a_M & \text{si } a_n > a_M \\ a_n - a_m & \text{si } a_n < a_m \\ 0 & \text{sinon} \end{cases} \quad (15)$$

Enfin, pour obtenir la loi de commande, la matrice d'interaction associée à cette information visuelle doit être estimée. En s'appuyant sur l'équation (13), la dérivée temporelle de $e_{\nabla a_n}$ est :

$$\dot{e}_{\nabla a_n} = \frac{\partial e_{\nabla a_n}}{\partial a_n} \frac{da_n}{dt} = \frac{\partial^2 \mathcal{V}_{a_n}}{\partial a_n^2} \mathbf{L}_{a_n} \mathbf{v} = \mathbf{L}_{\nabla a_n} \mathbf{v}, \quad (16)$$

où \mathbf{L}_{a_n} est la matrice d'interaction associée à a_n et $\mathbf{L}_{\nabla a_n}$ celle associée à $\nabla_{a_n} \mathcal{V}_{a_n}$. En utilisant l'approximation proposée dans (15), on obtient :

$$\frac{\partial^2 \mathcal{V}_{a_n}}{\partial a_n^2} = \begin{cases} 1 & \text{si } a_n < a_m, \text{ ou } a_n > a_M \\ 0 & \text{sinon.} \end{cases} \quad (17)$$

Puisque $e_{\nabla a_n} = 0$ pour $a_m < a_n < a_M$, \mathbf{L}_{a_n} peut être choisie comme une bonne approximation de $\mathbf{L}_{\nabla a_n}$. Une approximation de cette matrice d'interaction \mathbf{L}_{a_n} est donnée par [Tahri and Chaumette, 2005] :

$$\mathbf{L}_{a_n} = \begin{bmatrix} 0 & 0 & -\frac{1}{Z^*} & -a_n \varepsilon_1 & a_n \varepsilon_2 & 0 \end{bmatrix}, \quad (18)$$

avec :

$$\begin{aligned} \varepsilon_1 &= y_g + (y_g \mu_{02} + x_g \mu_{11} + \mu_{21} + \mu_{03}) / a \\ \varepsilon_2 &= x_g + (x_g \mu_{20} + y_g \mu_{11} + \mu_{12} + \mu_{30}) / a, \end{aligned}$$

où ε_1 et ε_2 peuvent être négligés par rapport à 1. Cette approximation est valide lorsque la caméra est parallèle à l'objet observé supposé plan (à une distance Z^*). Les expériences de la section 4 montreront que cette approximation permet tout de même d'obtenir des résultats satisfaisants (expériences dans lesquelles nous avons fixé $Z^* = 1$).

3.3.3. Contrôle de la position des projections

La fonction suivante contrôle les projections sur le plan image des points de l'ensemble d'intérêt. Il semble clair que l'ensemble de ces primitives doivent se projeter dans le champ de vision de la caméra.

Ainsi, les coordonnées de projection d'un point donné $\mathbf{x}_j = (x_j, y_j)$ sont considérées satisfaisantes si elles sont telles que : $x_j \in [x_m + \alpha; x_M - \alpha]$ et $y_j \in [y_m + \alpha; y_M - \alpha]$, où x_m, x_M, y_m et y_M sont les bords de l'image, et α un scalaire positif définissant une zone libre de projection \mathcal{I}_{libre} sur le plan image (voir la Figure 13).

La fonction \mathcal{V}_s caractérisant la projection des points est définie comme suit :

$$\mathcal{V}_s = \sum_j \mathcal{V}_{s(x_j)} \quad \text{avec} \quad \mathcal{V}_{s(x_j)} = g(x_m - x_j) + g(x_j - x_M) \\ + g(y_m - y_j) + g(y_j - y_M),$$

où $g(x)$ a déjà été donné dans (12). La Figure 14 trace cette fonction de coût dans le cas d'un seul point, ainsi qu'une des composantes de son gradient. $\nabla_s^T \mathcal{V}_s$ regroupe ici les gradients associés à chacun des points de l'ensemble d'intérêt \mathcal{M}_i . Il est donné par :

$$\nabla_s^T \mathcal{V}_s = (\nabla_s^T \mathcal{V}_s(\mathbf{x}_1), \dots, \nabla_s^T \mathcal{V}_s(\mathbf{x}_n)),$$

avec $n = \text{card}(\mathcal{M}_j)$ et :

$$\nabla_s^T \mathcal{V}_s(\mathbf{x}_j) = \begin{bmatrix} (x_j - x_M)h(x_j - x_M) + (x_j - x_m)h(x_m - x_j) + \mathcal{O}(x_j - x_M) - \mathcal{O}(x_m - x_j) \\ (y_j - y_M)h(y_j - y_M) + (y_j - y_m)h(y_m - y_j) + \mathcal{O}(y_j - y_M) - \mathcal{O}(y_m - y_j) \end{bmatrix}.$$

En utilisant la même approximation que précédemment, la matrice d'interaction de $\nabla_s^T \mathcal{V}_s$ est approchée par la matrice d'interaction \mathbf{L}_s associée aux coordonnées image des projections. Pour un point, cette matrice est de la forme :

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

où d_{i+1} est la distance entre le plan image de la vue ψ_{i+1} et le plan de référence π . $\mathbf{S} = (\mathbf{S}_1, \dots, \mathbf{S}_n)$ et $\mathbf{Q} = (\mathbf{Q}_1, \dots, \mathbf{Q}_n)$ sont deux matrices $2n \times 3$ qui ne dépendent pas de d_{i+1} :

$$\mathbf{S}_j = \frac{1}{\rho_j} \begin{bmatrix} -1 & 0 & x_j \\ 0 & -1 & y_j \end{bmatrix}$$

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

Le scalaire ρ_j est donné par l'équation (6), en utilisant l'homographie ${}^t\mathbf{H}_{i+1}$ entre la vue courante ψ_i et l'image ψ_{i+1} du chemin.

3.3.4. Mesure basée sur l'orientation 3D

Cette dernière mesure visuelle concerne la différence d'orientation mesurée entre la pose courante de la caméra et celles des images du chemin. Cette rotation peut être extraite de la matrice

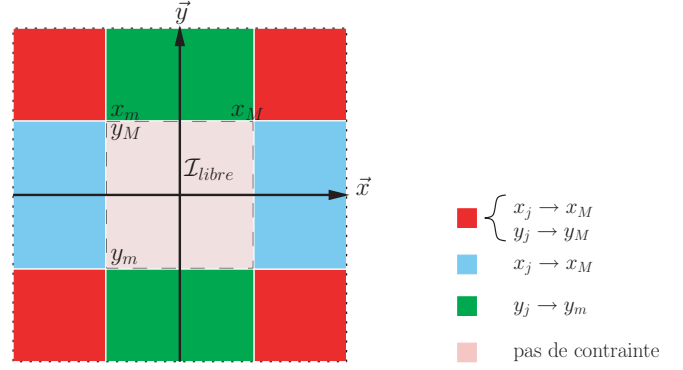


Figure 13. Zones d'activation déduite de la loi de commande tracées sur le plan image. La zone \mathcal{I}_{libre} est une restriction du cadre de l'image.

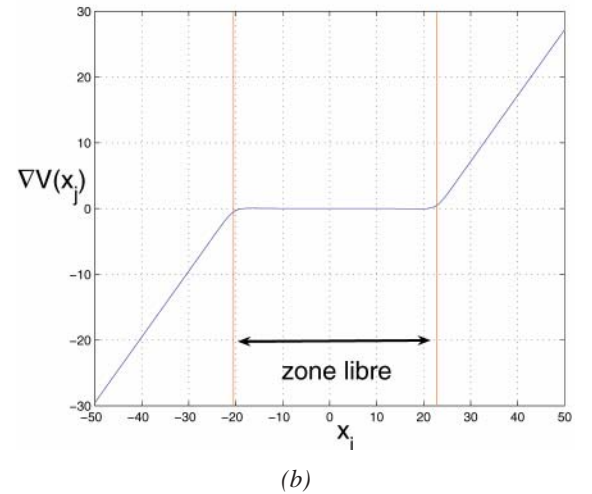
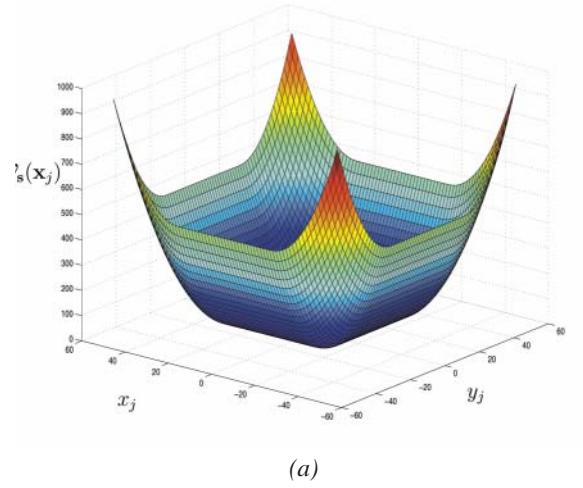


Figure 14. Fonction associée à la visibilité des points : (a) fonction pour un point, (b) gradient pour la coordonnée x_j .

d'homographie ${}^t\mathbf{H}_{i+1}$ reliant la vue courante ψ_i avec l'image ψ_{i+1} .

La représentation minimale de la rotation $\theta \mathbf{u}$ est obtenue à partir des coefficients $r_{ij}(i=1..3, j=1..3)$ de ${}^t\mathbf{R}_{i+1}$, en utilisant :

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

avec $\theta = \arccos((r_{11} + r_{22} + r_{33} - 1)/2)$, et où le sinus cardinal $\operatorname{sinc} \theta$ est tel que $\sin \theta = \theta \operatorname{sinc} \theta$.

Encore une fois, un intervalle définit la qualité de l'orientation courante de la caméra :

$$-p_\theta < \theta u_i < p_\theta,$$

où p est un scalaire positif appartenant à $[0 \ 1]$. La fonction de coût associée est :

$$\mathcal{V}_{\theta \mathbf{u}}(\theta u_i) = g(\theta u_i - p_\theta) + g(-p_\theta - \theta u_i), \quad (19)$$

dont le gradient est :

$$\begin{aligned} \nabla_{\theta \mathbf{u}} \mathcal{V}_{\theta \mathbf{u}}(\theta u_i) &= (\theta u_i - p_\theta) h(\theta u_i - p_\theta) + \mathcal{O}(\theta u_i - p_\theta) \\ &+ (\theta u_i + p_\theta) h(-p_\theta - \theta u_i) - \mathcal{O}(\theta u_i + p_\theta) \end{aligned} \quad (20)$$

La matrice d'interaction de $\nabla_{\theta \mathbf{u}}^\top \mathcal{V}_{\theta \mathbf{u}}$ est approximée par $\mathbf{L}_{\theta \mathbf{u}}$ [Malis and Chaumette, 2000] :

$$\mathbf{L}_{\theta \mathbf{u}} = [\mathbf{0}_3 \ \mathbf{L}_w],$$

$$\text{où } \mathbf{L}_w = \mathbf{I}_3 - \frac{\theta}{2} [\mathbf{u}]_\times + \left(1 - \frac{\operatorname{sinc} \theta}{\operatorname{sinc}^2 \frac{\theta}{2}} \right) [\mathbf{u}]_\times^2 \quad (21)$$

La fonction $\mathcal{V}_{\theta \mathbf{u}}$ est très similaire à celles définies précédemment. Notons que dans nos expériences, cette fonction n'est utilisée que pour contrôler les rotations autour des axes \vec{x} et \vec{y} . En effet, les rotations autour de l'axe optique ne permettent ni d'améliorer les conditions d'observation des amers visuels, ni de rapprocher le système robotique de sa position désirée. Mais il serait naturellement possible de contrôler ce degré de liberté pour d'autres applications.

La section suivante présente comment ces différentes mesures visuelles sont combinées pour calculer les mouvements du robot.

3.3.5. Loi de commande

Nous avons défini trois fonctions de coût décrivant chacune une contrainte différente sur la projection des points d'intérêt. Le système robotique doit donc se déplacer pour satisfaire chacune de ces contraintes. Ceci est réalisé en considérant simultanément les trois mesures visuelles dans la loi de commande :

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

où \mathbf{L} et ∇ sont respectivement un empilement des matrices d'interaction et des gradients des trois fonctions :

$$\mathbf{L} = (\mathbf{L}_s, \mathbf{L}_{a_n}, \mathbf{L}_{\theta \mathbf{u}}), \text{ et } \nabla = (\nabla_s^\top \mathcal{V}_s, \nabla_{a_n}^\top \mathcal{V}_{a_n}, \nabla_{\theta \mathbf{u}}^\top \mathcal{V}_{\theta \mathbf{u}})$$

Le gradient $\nabla_s \mathcal{V}_s$ est calculé comme décrit dans la Section 3.3.3. Les primitives \mathbf{x}_{n_j} utilisées appartiennent à l'ensemble \mathcal{M}_i qui regroupe les points appariés entre les vues ψ_i et ψ_{i+1} du che-

min. Leurs coordonnées sont obtenues soit par la phase de suivi, soit par transfert d'images. Dans $\nabla_{a_n} \mathcal{V}_{a_n}$ (défini dans la section 3.3.2), la mesure désirée a^* est calculée à partir des projections de points sur l'image ψ_{i+1} du chemin. Encore une fois, seules les primitives de l'ensemble \mathcal{M}_i sont utilisées pour calculer cette grandeur. Notons que dans nos expérimentations, un lissage sur trois itérations est réalisé. Ceci permet notamment d'assurer une meilleure continuité de la commande lors d'un changement d'ensemble d'intérêt.

Lorsque suffisamment de points du dernier ensemble \mathcal{M}_{N-1} sont visibles, le système robotique se situe dans le voisinage de sa position désirée. Un asservissement visuel classique est alors employé afin de réaliser la convergence finale.

4. Résultats expérimentaux

Des résultats expérimentaux sont maintenant présentés. Dans un premier temps, afin de pouvoir étudier le comportement de la loi de commande tout en s'affranchissant du bruit potentiel apporté par le suivi de points et le transfert d'images, les expériences sont réalisées en simulation. Les premiers résultats concernent un système robotique possédant cinq degrés de liberté (translations suivant \vec{x} , \vec{y} et \vec{z} , rotations autour de \vec{x} et \vec{y}). Puis le cas d'une caméra se déplaçant sur un plan est étudié. Enfin des expériences réalisées avec un bras articulé, et dans le cas d'une scène plane sont détaillées.

4.1. Système doté de cinq degrés de liberté

Nous présentons tout d'abord l'objet autour duquel la caméra doit se mouvoir. Il est illustré sur la Figure 15. Des points d'intérêt sont définis sur cet objet. Sur la Figure 17 est présenté un chemin d'images extraites de la base. Les positions associées à ces images sont illustrées sur la Figure 16.

La position de la caméra durant la navigation est présentée sur les Figures 18 et 19. Sur cette dernière figure, les lignes verticales correspondent à un changement d'ensemble d'intérêt \mathcal{M}_i . Les croix indiquent la position de la prochaine image du chemin juste avant le changement d'ensemble d'intérêt. Si le système convergait vers chacune de ces positions, toutes ces croix devraient être atteintes, ce qui n'est pas le cas.

Les premiers mouvements du robot rapprochent la caméra de l'objet, au moyen d'une translation le long de l'axe optique, grâce à \mathcal{V}_{a_n} . Cependant, l'objet étant proche du bord gauche de l'image, \mathcal{V}_s permet de compenser les translations sur \vec{z} par des translations sur \vec{x} ainsi que des rotations autour de \vec{y} . Cette dernière mesure, combinée à $\mathcal{V}_{\theta \mathbf{u}}$ assure ensuite le passage sur le dessus de l'objet, des itérations 370 à 1000. Puis des translations le long de \vec{x} générées par \mathcal{V}_s permettent de se rapprocher de la position désirée.

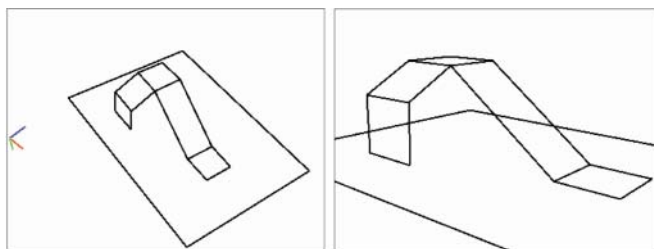


Figure 15. Vues de l'objet 3d utilisé pour les simulations.
Le repère de la caméra est défini par l'axe \vec{x} en rouge,
 \vec{y} en vert, et \vec{z} en bleu.

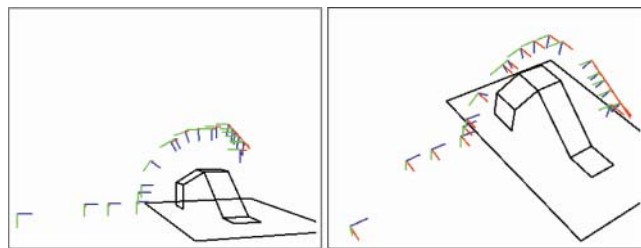


Figure 16. Exp. 1 : Positions associées au chemin d'images.

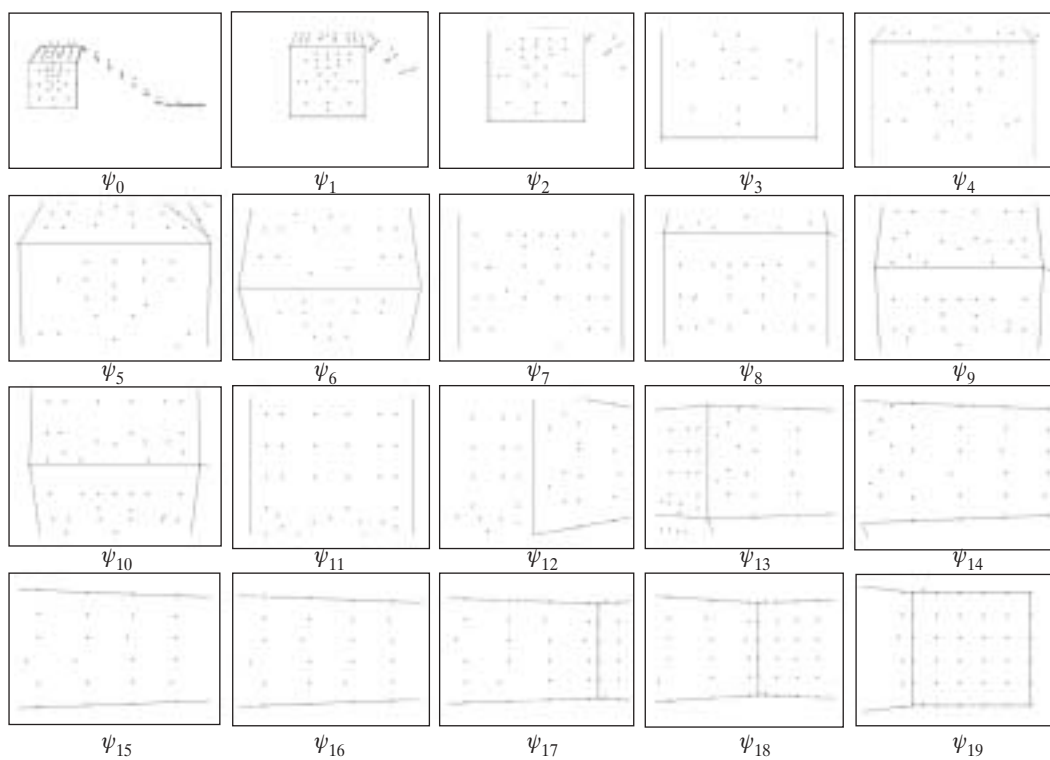


Figure 17. Exp. 1 : chemin d'images (ψ_0 : image initiale, ψ_{19} : image désirée).

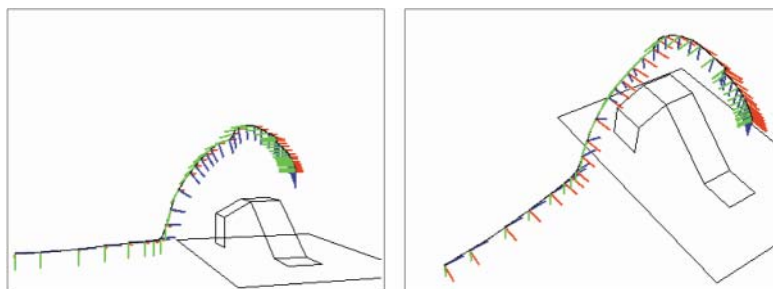


Figure 18. Exp. 1 : trajectoire réalisée.

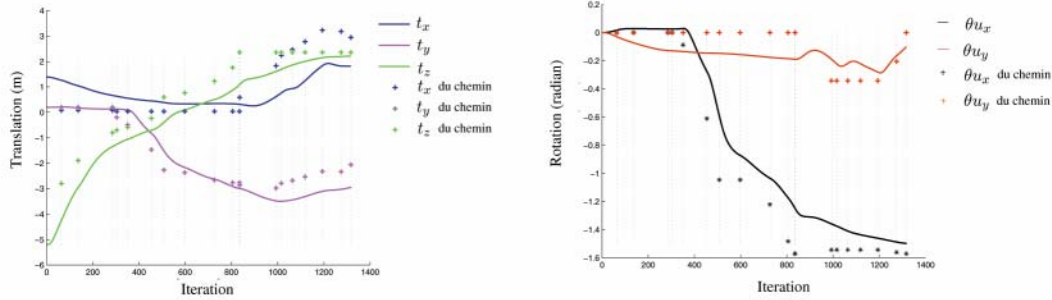


Figure 19. Exp. 1 : position de la caméra durant la navigation.

4.2. Système se déplaçant sur un plan

Dans l'expérience suivante, la loi de commande a été adaptée pour contrôler un système robotique se déplaçant sur un plan. L'environnement de navigation correspond à un couloir qui est défini par un ensemble de plans. Le robot est contrôlé par deux entrées : une pour les translations le long de l'axe \vec{z} , et une pour les rotations autour de l'axe \vec{y} . Les matrices d'interaction \mathbf{L}_s , $\mathbf{L}_{\theta u}$ et \mathbf{L}_{a_n} sont donc simplifiées en conséquence pour ne considérer que ces 2 degrés de liberté.

La Figure 20(a) présente le chemin d'images. Sur le début du chemin, on peut noter que certaines vues ne sont pas situées sur le plus court chemin. Les images ψ_1 et ψ_3 sont en effet décalées sur la gauche. Il est ici clair que le robot n'a pas besoin de converger vers ces vues pour atteindre sa position désirée.

Durant la phase de navigation, un bruit uniforme de cinq pixels a été appliqué sur les coordonnées des points d'intérêt, afin de simuler les imprécisions que peut engendrer l'algorithme de traitement d'images. Les figures 20(b) et 21 décrivent la trajectoire réalisée par le système robotique. De nouveau, on peut observer que le robot ne converge pas vers les poses associées aux vues ψ_1 et ψ_3 . En effet, sur la Figure 21, la ligne bleue (position du robot suivant l'axe \vec{x}) reste éloignée des croix 1 et 3. La robustesse au bruit de l'approche sur cette expérience s'explique par la propriété qualitative de la commande, qui, en étendant la zone de convergence, permet d'être moins sensible aux erreurs locales (comme une mauvaise estimation des coordonnées de points).

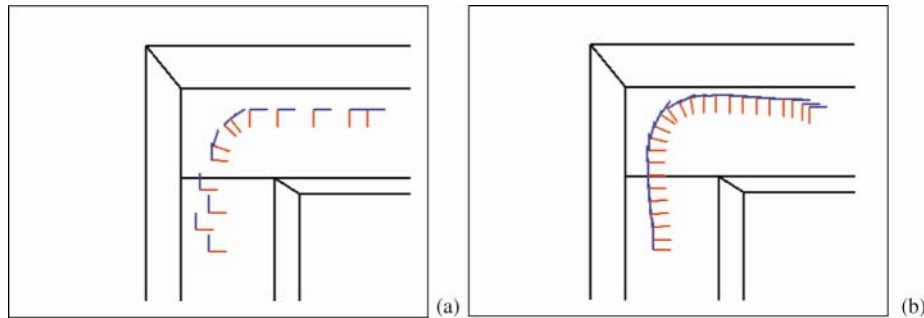


Figure 20. Exp. 2 : position des images du chemin, et trajectoire réalisée.

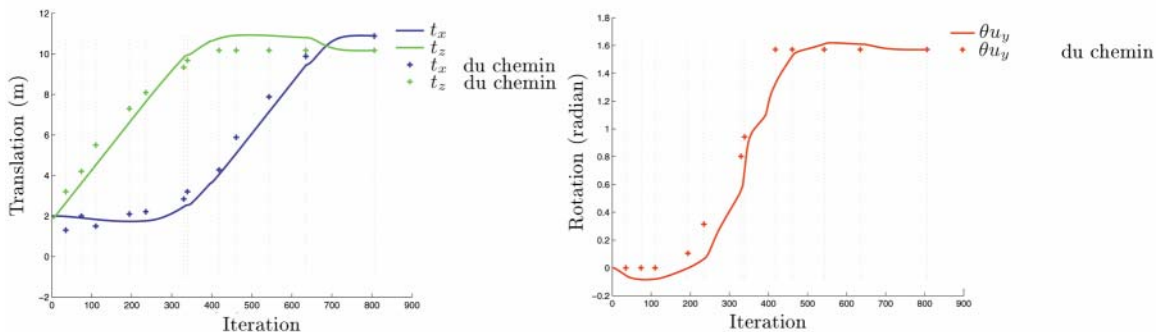


Figure 21. Exp. 2 : position et orientation du système durant la navigation.

4.3. Expérience sur un bras articulé

L'expérience suivante a été réalisée sur un bras articulé à six degrés de liberté et doté d'une caméra embarquée sur son effecteur. L'espace de navigation correspond à un plan sur lequel a été collé un ensemble de photos. Pour démontrer la validité de l'approche, le chemin considéré correspond à un cas où le robot ne peut pas se déplacer directement de la position initiale vers la position désirée. Les images extraites de la base qui définissent le chemin sont présentées sur la Figure 22.

Lorsque l'on considère une scène plane, le processus de transfert de points présenté dans la Section 3.2 se voit simplifié. Dans ce cas, la relation entre les projetés de même points 3D sur deux images se réduit à ${}^2\mathbf{x}_p \propto {}^2\mathbf{H}_{p_1} {}^1\mathbf{x}_p$, puisque tous les points appartiennent au plan de référence de la matrice d'homographie. Ainsi, l'estimation de cette homographie est suffisante pour réaliser le transfert. En effet, si l'on suppose que les correspondances entre les vues ψ_i et ψ_j sont assez nombreuses pour estimer l'homographie associée (quatre appariements suffisent), et si l'homographie entre les images du chemin ψ_i et ψ_j a été calculée au préalable, alors on peut déduire l'homographie entre les vues ψ_i et ψ_j , qui n'est rien d'autre que :

$${}^t\mathbf{H}_{p_j} = {}^t\mathbf{H}_{p_i} {}^i\mathbf{H}_{p_j} \quad (22)$$

La transfert de point s'appuie sur ces relations, et permet de détecter l'apparition de points d'intérêt dans le champ de vision de la caméra. Ces principes ont permis de créer la Figure 23(a), dans laquelle l'ensemble des points d'intérêt utilisés et l'ensemble des cadre des images ont été projetés sur le plan image associée à la première vue. On peut d'ailleurs remarquer que

tous les points ne sont pas visibles dans la première vue, et leur position ne permet clairement pas au système de se déplacer directement de sa position initiale vers la position désirée.

La Figure 23(b) présente la trajectoire réalisée par le point principal de la caméra durant la navigation. Déplacer la caméra afin de rendre visible les points de l'ensemble d'intérêt courant \mathcal{M}_i est suffisant pour parvenir à réaliser la tâche de navigation. La Figure 24 compare la trajectoire obtenue avec celles réalisées avec deux autres approches. La première consiste en un ensemble d'asservissements visuels 2D où le système converge successivement vers chacune des images du chemin. Une fois que l'erreur mesurée entre la position courante et désirée des points est suffisamment faible, le robot considère alors la prochaine image du chemin comme sa nouvelle position désirée. Dans la deuxième approche [Mezouar *et al.*, 2002], le robot converge toujours vers les images intermédiaires avec un asservissement visuel 2D, mais l'asservissement courant est stoppé lorsqu'assez de primitives du prochain ensemble sont dans le champ de vision de la caméra (information obtenue par transfert de point via la relation (22)). La prochaine image du chemin est alors considérée comme la nouvelle position désirée. Ainsi, le robot ne converge plus totalement vers toutes les images du chemin, comme on peut le noter sur la Figure 24, mais est tout de même encore dépendant des poses des images intermédiaires. Avec la méthode proposée, comme on peut le voir, tout en assurant que le robot reste dans des zones où assez de points sont visibles, le robot parvient à se diriger vers la position désirée. Dans la dernière expérience présentée, une rotation de 80 degrés est appliquée aux images ψ_1 et ψ_5 du chemin. La réalisation de ce chemin par la première approche force le robot à réaliser ces rotations inutiles $\psi_0\psi_1$, $\psi_1\psi_2$, $\psi_4\psi_5$ et $\psi_5\psi_6$. La deuxième



Figure 22. Exp. 3: chemin d'images. ψ_0 : image initiale, ψ_6 : image désirée. Les autres vues ont été extraites automatiquement de la base.

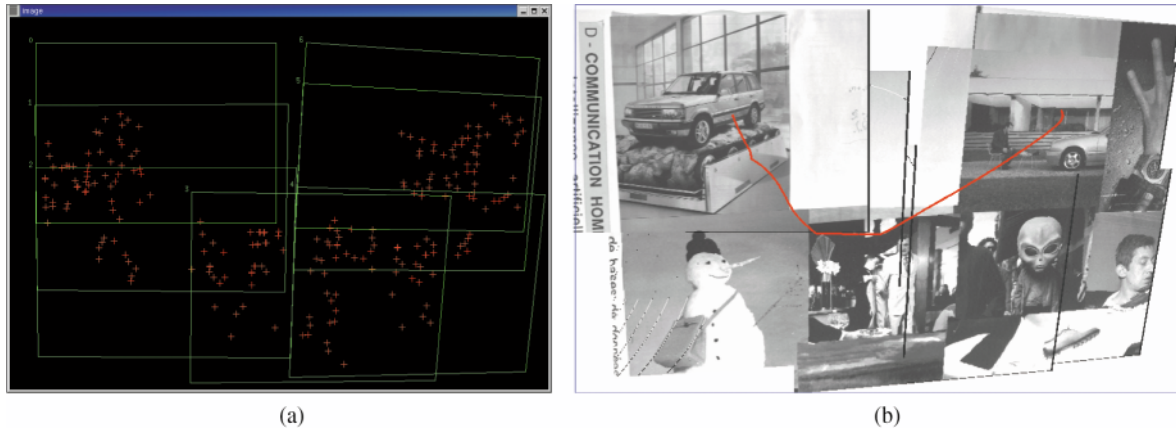


Figure 23. Exp. 3: (a) points et cadre des images projetés sur le premier plan image
(b) trajectoire du point principal de la caméra projeté sur le premier plan image.

méthode, bien qu'elle s'affranchisse d'une convergence totale vers les images intermédiaires, réalise tout de même une partie de ces rotations. La Figure 25 compare la trajectoire obtenue avec notre approche pour le chemin sans rotation et celui obtenu

quand ψ_1 et ψ_5 subissent une rotation. Les deux trajectoires sont quasiment identiques. Les rotations autour de l'axe optique n'affectent donc pas notre approche.

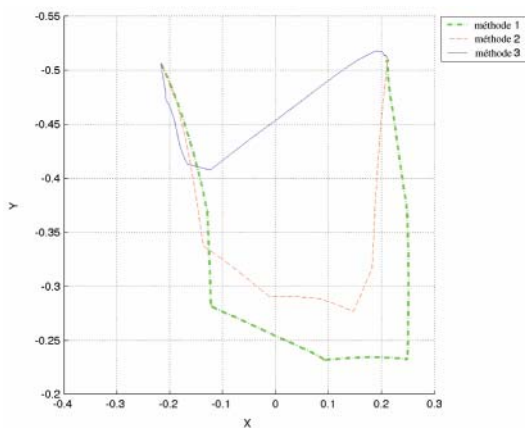


Figure 24. Exp. 3: Comparaison des trajectoires 2d de la caméra pour les trois approches.

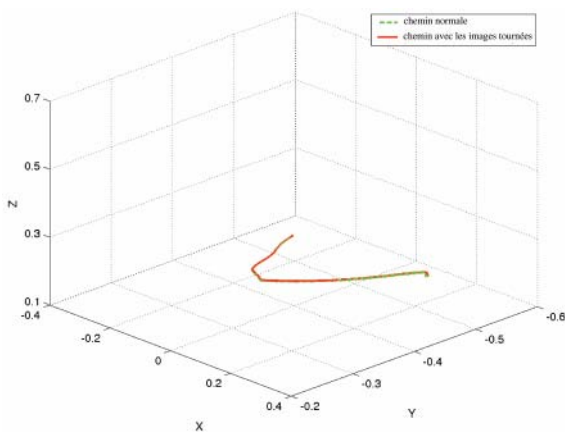


Figure 25. Exp. 3: Comparaison des trajectoires du robot (pour le chemin défini sur la figure 22 et le même avec les images tournées.)

5. Conclusion

Cet article a proposé une nouvelle loi de commande pour la réalisation de tâches de navigation basées vision. Un chemin d'images est tout d'abord extrait de la mémoire visuelle du robot qui décrit l'environnement de navigation. Ce chemin d'images définit les primitives visuelles, des points, que doit observer la caméra durant sa navigation. La loi de commande proposée ne nécessite pas une reconstruction 3D de l'environnement. De plus, les images du chemin ne sont pas considérées comme un ensemble de cibles intermédiaires. Les mouvements du robot sont définis en s'appuyant sur les points d'intérêt apparés entre les images successives du chemin. Ces ensembles constituent des descriptions des différentes zones que doit traverser le système robotique. En imposant au robot d'observer ces points dans de bonnes conditions, celui-ci parvient à se déplacer progressivement vers sa position désirée.

Un asservissement visuel qualitatif a été proposé. Celui-ci est basé sur un ensemble de fonctions de coût, caractérisant les positions des projections des points. L'originalité de ce formalisme est qu'il ne s'appuie ni sur une position désirée de la caméra ni sur une position désirée des projetés des points, mais sur des intervalles de valeurs jugées satisfaisants. Les expériences présentées, réalisées en simulation et sur un système réel, montrent le bien fondé de notre approche.

L'étape suivante de ces recherches concerne l'application du formalisme sur un véhicule robotisé. Cela impose de définir notamment des mesures visuelles adaptées aux types de mouvement que peuvent réaliser ces systèmes. De plus, nous travaillons sur l'extension de la loi de commande afin de pouvoir prendre en compte les contraintes de non-holonomie qui sont caractéristiques de ce type de véhicule.

Enfin, l'application grandeur nature d'une telle approche impose de prendre en considération le problème de stockage et de mise à disposition des informations visuelles utiles lors de la navigation. Le stockage dans le véhicule autonome de l'ensemble de la description de l'environnement limite fortement la dimension des espaces de navigation pouvant être considérés. L'un des avantages de l'approche proposée est que le système robotique n'a pas besoin pour naviguer de la description de l'ensemble de son environnement, mais simplement du chemin d'images décrivant le chemin qu'il doit parcourir. De ce fait, le problème du stockage pourrait être traité par une approche client-serveur, dans laquelle le serveur aurait pour tâche de définir le chemin d'images à réaliser, et fournirait au véhicule client simplement le chemin d'images pour la réalisation du parcours. De telles approches ont été considérées avec attention au sein des projets nationaux ROBEA BODEGA et PREDIT MOBIVIP dans lesquels s'inscrivent les travaux présentés dans cet article.

Références

- [Argyros *et al.*, 2001] ARGYROS A., BEKRIS C. and ORPHANOUDA-KIS S. (2001). Robot homing based on corner tracking in a sequence of panoramic views. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3-10, Kauai, USA.
- [Berrani *et al.*, 2003] BERRANI S., AMSALEG L. and GROS P. (2003). Approximate searches: k-neighbors+precision. In *ACM Int. Conf. on Information and Knowledge Management*, pp. 70-77, New Orleans, USA.
- [Blanc *et al.*, 2005] BLANC G., MEZOUAR Y. and MARTINET P. (2005). Indoor navigation of a wheeled mobile robot along visual routes. In *IEEE Int. Conf. on Robotics and Automation*, Barcelone, Espagne.
- [Burschka and Hager, 2001] BURSCHKA D. and HAGER G.D. (2001). Vision-based control of mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, pp. 1707-1713, Séoul, Corée du Sud.
- [Cobzas *et al.*, 2003] COBZAS D., ZHANG H. and JAGERSAND M. (2003). Image-based localization with depth-enhanced image map. In *IEEE Int. Conf. on Robotics and Automation*, pp. 1570-1575, Taïpeh, Taiwan.
- [Dao *et al.*, 2003] DAO N. X., YOU B. J., OH S. R. and HWANGBO M. (2003). Visual self-localization for indoor mobile robots using natural lines. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 1252-1255, Las Vegas, USA.
- [De La Torre and Black, 2001] DE LA TORRE F. and BLACK M. J. (2001). Robust principal component analysis for computer vision. In *IEEE Int. Conf. on Computer Vision*, volume 1, pp. 362-369, Vancouver, Canada.
- [Espiau *et al.*, 1992] ESPIAU B. and CHAUMETTE F. and RIVES P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. s. on Robotics and Automation*, 8(3) :313-326.
- [Faugeras and Lustman, 1988] FAUGERAS O. and LUSTMAN F. (1988). Motion and structure from motion in a piecewise planar environment. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2 :485-508.
- [Harris and Stephens, 1988] HARRIS C. and STEPHENS M. (1988). A combined corner and edge detector. In *Alvey Vision Conf.*, pp. 147-151, Université de Manchester, Angleterre.
- [Hartley and Zisserman, 2000] HARTLEY R. and ZISSERMAN A. (2000). *Multiple view geometry in computer vision*. Cambridge University Press, Angleterre.
- [Hayet *et al.*, 2002] HAYET J.B., LERASLE F. and DEVY M. (2002). A Visual Landmark Framework for Indoor Mobile Robot Navigation. In *IEEE Int. Conf. on Robotics and Automation*, pp. 3942-3947, Washington, USA.
- [Jin *et al.*, 2001] JIN H., FAVARO P. and SOATTO S. (2001). Real-time feature tracking and outlier rejection with changes in illumination. In *IEEE Int. Conf. on Computer Vision*, volume 1, pp. 684-689, Vancouver, Canada.
- [Jones *et al.*, 1997] JONES S., ANDERSEN C. and CROWLEY J.L. (1997). Appearance based processes for visual navigation. In *IEEE Int. Conf. on Intelligent Robots and Systems*, volume 2, pp. 551-557.
- [Koščekà *et al.*, 2003] KOŠECKÀ J., ZOUH L., BARBER P. and DURIC Z. (2003). Qualitative image based localization in indoor environments. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3-10, Madison, USA.
- [Latombe, 1993] LATOMBE J. (1993). *Robot Motion Planning*. Kluwer.
- [Laumond, 1998] LAUMOND J. (1998). *Robot Motion Planning and Control*. Lectures Notes in Control and Information Sciences, Springer-Verlag.
- [Lowe, 2004] LOWE D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2) :91-110.
- [Malis and Chaumette, 2000] MALIS E. and CHAUMETTE F. (2000). 2 1/2d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *Int. Journal of Computer Vision*, 37(1) :79-97.
- [Matsumoto *et al.*, 2000] MATSUMOTO Y., INABA M. and INOUE H. (2000). View-based approach to robot navigation. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 1702-1708, Takamatsu, Japon.
- [Mezouar *et al.*, 2002] MEZOUAR Y., REMAZEILLES A., GROS P. and CHAUMETTE F. (2002). Image interpolation for image-based control under large displacement. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pp. 3787-3789, Washington, USA.
- [Rasmussen and Hager, 1996] RASMUSSEN C. and HAGER G. (1996). Robot navigation using image sequences. In *National Conf. on Artificial Intelligence*, volume 2, pp. 938-943, Portland, USA.
- [Remazeilles *et al.*, 2004] REMAZEILLES A., CHAUMETTE F. and GROS P. (2004). Robot motion control from a visual memory. In *IEEE Int. Conf. on Robotics and Automation*, volume 4, pp. 4695-4700, New Orleans, USA.
- [Royer *et al.*, 2004] ROYER E., LHUILLER M., DHOME M. and CHATEAU T. (2004). Towards an alternative gps sensor in dense urban environment from visual memory. In *British Machine Vision Conf.*, Londres, Angleterre.
- [Schmid and Mohr, 1997] SCHMID C. and MOHR R. (1997). Local gray-value invariants for image retrieval. *IEEE Trans. s on Pattern Analysis and Machine Intelligence*, 19(5) :530-534.
- [Se *et al.*, 2002] SE S., LOWE D. and LITTLE J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Int. Journal of Robotics Research*, 21(8) :735-758.
- [Shashua and Navab, 1996] SHASHUA A. and NAVAB N. (1996). Relative affine structure : Canonical model for 3d from 2d geometry and applications. *IEEE Trans.s on Pattern Analysis and Machine Intelligence*, 18(9) :873-833.
- [Shi and Tomasi, 1994] SHI J. and TOMASI C. (1994). Good features to track. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 593-600, Seattle, USA.
- [Smeulders *et al.*, 2000] SMEULDERS A. W. M., WORRING M. and SANTINI S., GUPTA A. and JAIN R. (2000). Content-based image retrieval at the end of the early years. *IEEE Trans.s on Pattern Analysis and Machine Intelligence*, 22(12) :1349-1380.
- [Tahri and Chaumette, 2005] TAHRI O. and CHAUMETTE F. (2005). Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans.s on Robotics*, 21(6) :1116-1127.
- [Thrun *et al.*, 2000] THRUN S., BURGARD W. and FOX D. (2000). A real time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *IEEE Int. Conf. on Robotics and Automation*, pp. 321-328, San Francisco USA.
- [Zhou *et al.*, 2003] ZHOU C., WEI Y. and TAN T. (2003). Mobil robot self-localization based on global visual appearance features. In *IEEE Int. Conf. on Robotics and Automation*, pp. 1271-1276, Taïpeh, Taiwan.



Anthony **Remazeilles**

Anthony Remazeilles a obtenu le diplôme d'Ingénieur en Informatique de l'Institut National des Sciences Appliquées de Rennes en 2001. En 2004, il a soutenu un doctorat en Informatique à l'Université de Rennes 1. En 2005 et 2006, il a occupé un poste d'Attaché Temporaire à l'Enseignement et à la Recherche, conjointement à l'INSA et à l'IRISA de Rennes. Il effectue actuellement un séjour post-doctoral au CEA-LIST, à Fontenay aux Roses. Ses travaux de recherche concernent la robotique et la vision par ordinateur.



François **Chaumette**

François Chaumette est né à Nantes en 1963. Il a soutenu à l'Université de Rennes 1 une thèse en informatique en 1990, puis une habilitation à diriger des recherches en 1998. Chercheur à l'IRISA depuis sa thèse, il y est directeur de recherches INRIA et responsable de l'équipe Lagadic. Son domaine de recherches porte sur l'asservissement visuel et la vision robotique.



Patrick **Gros**

Patrick Gros est directeur de recherche INRIA. Après avoir travaillé plusieurs années sur le problème de l'appariement et de l'indexation des images, il a fondé l'équipe Texmex à l'IRISA, à Rennes, qui s'occupe d'indexation multimédia. Le but de l'équipe est de développer des approches pluridisciplinaires (traitement d'images, traitement automatique des langues, bases de données, statistiques, analyse du son et de la parole) pour décrire et analyser des vidéos, et en particulier la télévision. Ses centres d'intérêt sont l'indexation multidimensionnelle et la description multimodale des documents multimédias.



