

N° d'ordre: 2586

THÈSE

présentée

DEVANT L'UNIVERSITÉ DE RENNES I

pour obtenir

le grade de : *DOCTEUR DE L'UNIVERSITÉ DE RENNES I*

Mention : Informatique

par

Youcef Mezouar

Équipe d'accueil: VISTA (IRISA/INRIA, RENNES)

École doctorale : Mathématiques, Informatique, Signal, Electronique et
Télécommunications

Composante Universitaire : Institut de formation supérieure
en informatique et communication

Titre de la thèse :

Planification de trajectoires pour l'asservissement visuel

Soutenue le 28 Novembre 2001, devant la commission d'examen

COMPOSITION DU JURY :

M.	Claude	LABIT	Président
MM.	Philippe Claude	MARTINET SAMSON	Rapporteurs
MM.	Michel Philippe François	de MATHELIN SOUERES CHAUMETTE	Examineurs

A mes parents

Remerciements

Ce travail a été réalisé à l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires, INRIA Rennes), au sein de l'équipe Vista.

Je tiens à remercier Claude Labit, Directeur de Recherche à l'IRISA Rennes, pour avoir présidé le jury.

Philippe Martinet, Professeur des Universités à l'IFMA Clermont-Ferrand, et Claude Samson, Directeur de Recherche à l'INRIA Sophia-Antipolis, pour avoir accepté d'être les rapporteurs de cette thèse. Leur regard critique et leur nombreux commentaires m'ont permis d'améliorer significativement la qualité du manuscrit.

Je remercie également Michel de Mathelin, Professeur des Universités à l'ENSP de Strasbourg, et Philippe Soueres, Chargé de Recherche au LAAS Toulouse, d'avoir examiné avec rigueur ce travail de thèse.

Je souhaite exprimer toute ma gratitude à François Chaumette, directeur de recherche à l'IRISA Rennes, pour sa disponibilité durant ces trois années de thèse qu'il a encadré avec rigueur tout en m'accordant une grande liberté.

Je voudrais enfin remercier tous ceux qui m'ont entouré durant ces trois années : les membres du projet VISTA que je ne citerai pas de peur d'en oublier, ma famille, Brigitte, Jackot, François, Rémi.

Enfin, un merci particulier à Marion qui a contribué pour moitié à mon travail par sa présence, sa patience et sa bonne humeur.

Table des matières

1	Introduction	7
1.1	Introduction générale	7
1.2	Robustesse aux erreurs de modélisation	8
1.3	Les trajectoires dans l'image	11
1.4	Les trajectoires dans l'espace de travail	12
1.5	Les trajectoires dans l'espace articulaire	13
1.6	Contributions	14
1.7	Organisation du manuscrit	14
I	Rappels et motivations	17
2	Géométrie d'un système muni d'une caméra	19
2.1	Espace projectif et transformation homogène	20
2.2	Modélisation de la caméra	22
2.3	Reconstruction projective et euclidienne	24
2.3.1	Les expressions matricielles de la contrainte épipolaire	25
2.3.2	Estimation de la géométrie épipolaire et du déplacement euclidien	28
2.4	Conclusion	32
3	L'asservissement visuel	35
3.1	Introduction	35
3.2	La commande en asservissement visuel	37
3.2.1	Un bref aperçu	37
3.2.2	La commande par fonction de tâche	38
3.2.2.1	Définition de la fonction de tâche	38
3.2.2.2	Réalisation d'une commande en vitesse	39
3.3	Choix de la consigne	42
3.4	Le choix de l'espace de contrôle	43
3.4.1	L'espace Image	43
3.4.2	L'espace Cartésien	46
3.4.3	L'espace Image et l'espace Cartésien	47

3.5	Conclusion	48
II	Planification de trajectoires	51
4	Introduction et état de l'art	53
4.1	Introduction	53
4.2	Génération de trajectoires en robotique	56
4.2.1	Les méthodes locales	57
4.2.2	Les méthodes globales	58
4.2.3	Les méthodes continues	59
4.2.4	Les méthodes probabilistes	60
4.3	La planification de trajectoires dans l'image	61
5	Génération de trajectoires à minimum d'énergie ou d'accélération	63
5.1	Etude du cas sans contrainte de visibilité	63
5.1.1	Mouvement de rotation pure	64
5.1.1.1	Quelques éléments sur le groupe des rotations	64
5.1.1.2	Trajectoire optimale de la colinéation	65
5.1.2	Mouvement de translation pure	69
5.1.3	Mouvement rigide	71
5.1.4	Les trajectoires dans l'image	75
5.1.5	Algorithmes et résultats de planification	76
5.1.5.1	Déplacement en rotation	76
5.1.5.2	Déplacement en translation	77
5.1.5.3	Déplacement en rotation et translation	77
5.1.5.4	Robustesse vis-à-vis des erreurs de calibration	86
5.1.6	Application à l'asservissement visuel	90
5.1.6.1	Suivi des trajectoires par asservissement visuel 2D	90
5.1.7	Résultats expérimentaux	93
5.1.7.1	Cible plane	94
5.1.7.2	Cible non-plane	102
5.1.7.3	Scène complexe	109
5.2	Application à l'interpolation de N images	112
5.2.1	Le problème	112
5.2.2	Trajectoire minimale	113
5.2.3	Résultats expérimentaux	114
5.3	Introduction de la contrainte de visibilité	119
5.3.1	Définition du problème	119
5.3.2	Exemple	120
5.4	Conclusion	124

6	Approche par la méthode des potentiels	125
6.1	La méthode des potentiels	126
6.1.1	L'approche classique	126
6.1.2	L'approche utilisée	127
6.2	Cas d'une cible connue	130
6.2.1	Trajectoire de la caméra dans \mathcal{W}	130
6.2.2	Trajectoire des primitives dans l'image	131
6.2.3	Trajectoire dans l'espace articulaire du robot	131
6.2.4	Champ de potentiel attractif	133
6.2.5	Champ répulsif relatif aux butées articulaires	133
6.2.6	Champ répulsif relatif à la contrainte de visibilité	135
6.2.7	Résumé	137
6.3	Cas d'un objet inconnu	137
6.3.1	Trajectoire partielle de la caméra	139
6.3.2	Trajectoire des primitives dans l'image	140
6.3.3	Influence du paramètre d^f	141
6.3.4	Influence de la calibration de la caméra	142
6.3.5	Résumé	143
6.4	Interpolation par des fonctions B-splines	143
6.5	Suivi des trajectoires par asservissement visuel 2D	146
6.6	Résultats expérimentaux	147
6.6.1	Conditions expérimentales	147
6.6.2	Cas d'une cible plane connue	148
6.6.3	Cas d'une cible plane inconnue	151
6.6.4	Résultats pour un objet non-planaire et inconnu	156
6.6.5	Evitement des butées articulaires	160
6.7	Conclusion	165
	 Conclusion	 167
	 Annexes	 173
A	Description d'une méthode numérique de résolution d'un problème de commande optimale	173
A.1	Introduction	173
A.2	Approche variationnelle	173
A.2.1	La commande optimale	174
A.2.2	Le problème de Bolza	177
A.2.3	Reformulation du problème de la commande optimale sous la forme de Bolza	178
A.2.4	Résolution numérique du problème de Bolza	180

B	Trajectoires optimales de la caméra	183
B.1	Problème PCN1	184
B.2	Problème PCN2	185
C	Les auto-occultations	187
C.1	Potentiel répulsif	187
C.2	Résultats expérimentaux	188
	Publications	209

Chapitre 1

Introduction

1.1 Introduction générale

La volonté des chercheurs de reproduire les capacités humaines de perception et d'action dans les systèmes robotisés a conduit à l'intégration de données issues de capteurs extéroceptifs, et plus particulièrement de celles issues d'une caméra. L'objectif affiché est que les capteurs visuels fournissent une information suffisamment riche pour permettre aux robots de réaliser, de manière autonome, des tâches dans des environnements partiellement connus, ou complètement inconnus. D'un point de vue méthodologique, l'asservissement visuel consiste à intégrer directement dans la boucle de commande des robots, des informations extraites des images fournies par des caméras afin de réaliser l'action souhaitée. En pratique, cela permet un élargissement important du domaine d'application de la robotique, et une amélioration considérable de la précision obtenue.

Les techniques d'asservissement visuel classiques sont efficaces pour une grande classe d'applications mais se heurtent à des difficultés, en particulier lorsque le déplacement à effectuer est très important. Les approches les plus anciennes sont basées sur la régulation à zéro de l'erreur entre les valeurs courante et désirée d'informations visuelles sélectionnées, soit dans l'image, soit dans l'espace 3-D. Des informations de type 3-D et 2-D peuvent également être combinées pour construire le signal d'erreur. Cette dernière approche, appelée dans la suite approche hybride ou 2D1/2, permet d'améliorer de manière significative le comportement obtenu [Chaumette 00a]. Toutes ces approches utilisent en général une consigne fixe et peuvent alors conduire à des comportements inadéquats. En effet, dans tous les cas de figures, en utilisant une consigne fixe, il est difficile d'introduire des contraintes sur les trajectoires réalisées et d'assurer la convergence quelle que soit la position initiale du robot. Les difficultés potentielles auxquelles nous sommes confrontés se situent tout d'abord au niveau des trajectoires du robot dans son espace de travail, mais aussi dans l'espace articulaire. Les primitives visuelles dans l'image peuvent également avoir des trajectoires non souhaitées les amenant à sortir du champ de vi-

sion de la caméra. Les problèmes de stabilité et de robustesse vis-à-vis des différentes erreurs sur les modèles utilisés deviennent également importants quand l'erreur entre la consigne et la mesure est grande. L'objet de ce chapitre introductif est de mettre en évidence, de manière qualitative et en donnant quelques exemples, ces difficultés potentielles qui constituent les motivations premières de ce travail de thèse (la description précise des différentes approches et de leurs difficultés respectives font l'objet du chapitre 3). Nous présenterons également nos principales contributions dans la section 1.6 et l'organisation du manuscrit dans la section 1.7.

1.2 Robustesse aux erreurs de modélisation

Le schéma qui consiste à planifier les trajectoires du système et à les suivre ensuite, est très intéressant du point de vue de la commande. Ce schéma nécessite uniquement une commande localement stable et robuste pour obtenir des résultats satisfaisants. Les commandes peuvent, par ailleurs, être linéarisées autour de la trajectoire nominale. En effet, en utilisant une consigne fixe, si la modélisation du système est parfaite, on peut construire un schéma de commande permettant d'obtenir le comportement souhaité. Si au contraire, les modèles utilisés sont erronés, les trajectoires obtenues peuvent s'avérer surprenantes et le système peut même diverger si les erreurs sont trop importantes (voir figure 1.1). En utilisant une consigne variable, le comportement du système peut être amélioré de manière significative (voir figure 1.2).

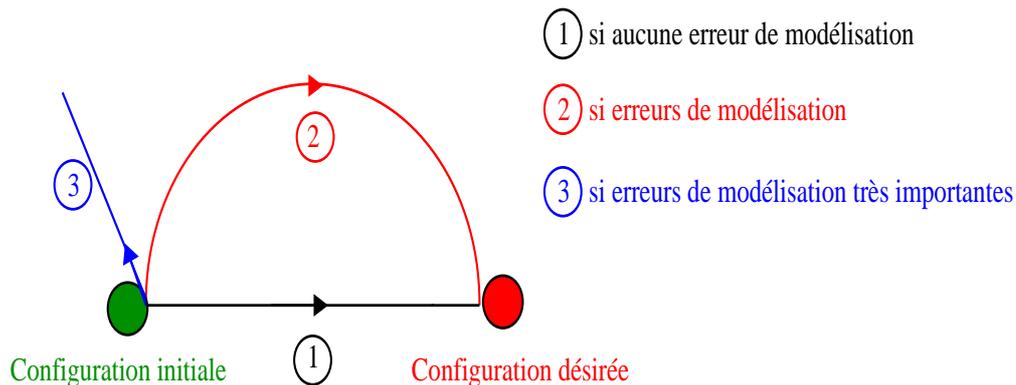


FIG. 1.1 – *Lorsqu'une consigne fixe est utilisée, les trajectoires obtenues peuvent s'avérer inadéquates en présence d'erreurs de modélisation*

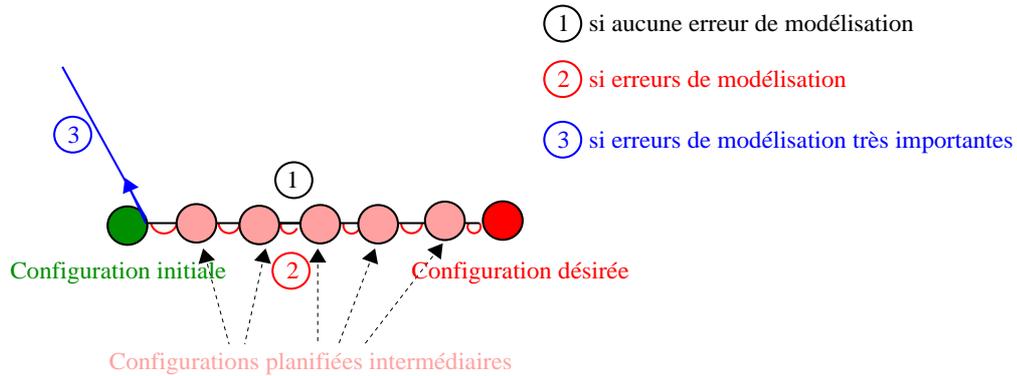


FIG. 1.2 – *En utilisant une consigne variable, le comportement du système peut être amélioré de manière significative*

Dans le contexte de l’asservissement visuel, cette approche est très pertinente puisque nous disposons de lois de commande connues pour être localement très stables et robustes. Nous en présentons ici une illustration. Nous souhaitons contrôler les mouvements d’un bras manipulateur sur lequel est montée une caméra observant un objet composé de quatre marques blanches, de telle sorte qu’à la fin de la tâche, l’image de la scène corresponde à une image apprise. Dans le cas présent, les images correspondant aux positions initiale et désirée sont illustrées sur les Figures 1.3(a) et 1.3(b). L’expérience montre la différence de trajectoire, obtenue d’une part, lorsque les paramètres de calibration de la caméra sont parfaitement connus ou lorsque ceux-ci sont entachés d’erreur et d’autre part, lorsque la consigne est fixe ou variable. Nous utilisons pour cette expérience une approche 2-D. Des résultats similaires sont obtenus avec une approche 3D. Afin de vérifier la robustesse vis-à-vis des paramètres de calibration, nous avons réalisé trois expérimentations :

1. Figure 1.4 : paramètres intrinsèques exacts et consigne fixe;
2. Figure 1.5 : paramètres intrinsèques entachés d’une erreur de 40% et consigne fixe;
3. Figure 1.6 : paramètres intrinsèques entachés d’une erreur de 40%. La consigne varie cette fois à chaque itération de telle sorte que les trajectoires désirées dans l’image soient des lignes droites.

Comme attendu, la trajectoire de chaque point d’intérêt dans l’image est une ligne droite lorsque les paramètres de calibration sont parfaitement connus (se référer à la Figure 1.4). Par contre, en présence d’erreurs de modélisation, nous notons que les trajectoires obtenues sont très éloignées de celles prévues (voir Figure 1.5). Dans ce cas, le mouvement des points dans l’image est imprévisible et une partie de l’objet d’intérêt pourrait même quitter l’image. Comme nous pouvons le constater sur la Figure 1.6, en utilisant une consigne variable, les trajectoires désirées sont effectivement réalisées, même en présence d’importantes erreurs de modélisation. Ce simple procédé nous a permis, dans ce cas, d’améliorer significativement la robustesse du système vis-à-vis des erreurs de modélisation.

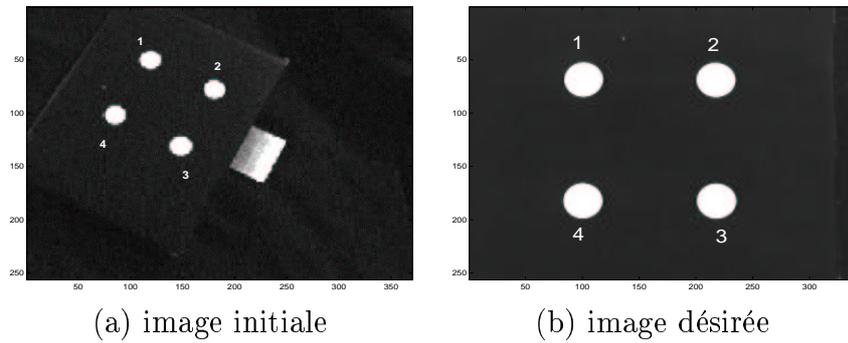


FIG. 1.3 – Images de l'objet d'intérêt

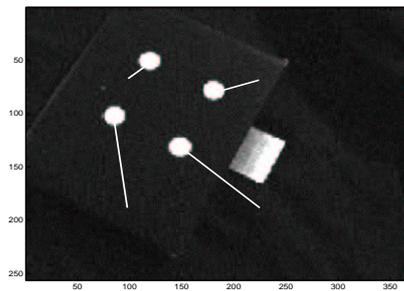


FIG. 1.4 – Trajectoires dans l'image : calibration parfaite et consigne fixe

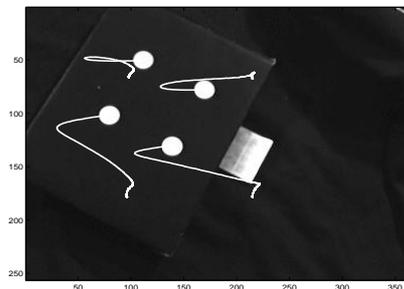


FIG. 1.5 – Trajectoires dans l'image : mauvaise calibration et consigne fixe

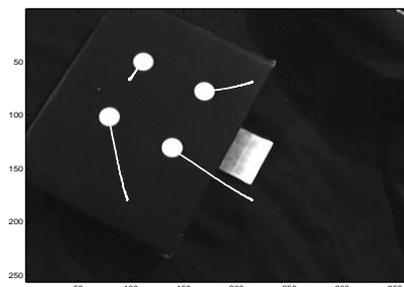


FIG. 1.6 – Trajectoires dans l'image : mauvaise calibration et consigne variable

1.3 Les trajectoires dans l'image

Les approches utilisant des informations 3-D pour synthétiser le signal d'erreur à réguler se heurtent à une difficulté majeure : les trajectoires dans l'image des primitives visuelles utilisées pour le calcul de la commande ne sont pas contrôlées. Ainsi, l'objet d'intérêt peut sortir du champ de vision de la caméra durant son déplacement et faire échouer l'asservissement. Nous présentons dans l'exemple suivant un tel cas de figure. Les images correspondant aux positions initiale et désirée sont illustrées sur les Figures 1.7(a) et 1.7(b). Le déplacement à réaliser est très important. Les trajectoires qu'effectueraient les quatre marques blanches en utilisant un asservissement 3-D, dans le cas où les modèles sont parfaitement connus, sont données sur la Figure 1.8. On observe que les primitives visuelles sortent largement du champ de vision de la caméra. L'asservissement visuel 3-D classique échouerait donc dans ce cas. L'introduction de primitives directement extraites de l'image dans les approches 2-D et 2-D-1/2 améliore significativement la trajectoire de l'objet d'intérêt dans l'image. Cependant, certaines parties de celui-ci peuvent éventuellement sortir du champ de vision de la caméra si les erreurs de modélisation et le déplacement à réaliser sont importants.

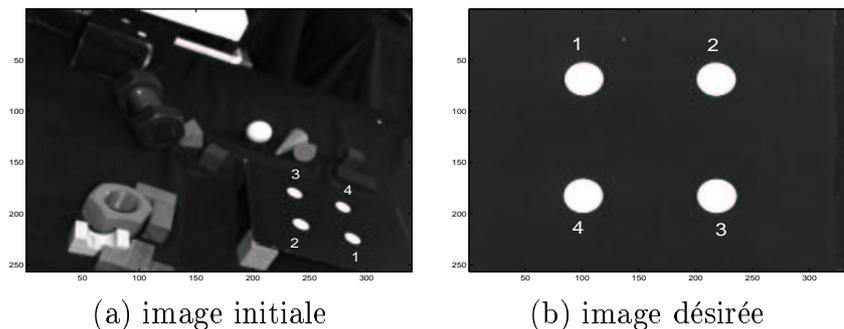


FIG. 1.7 – Images de l'objet d'intérêt

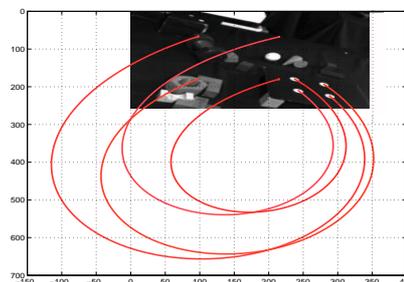


FIG. 1.8 – Trajectoires dans l'image : les points d'intérêt quittent l'image

1.4 Les trajectoires dans l'espace de travail

Les techniques d'asservissement visuel 2-D utilisent directement des informations extraites de l'image. Les lois de commande consistent alors à contrôler le mouvement de la caméra afin que les mesures dans l'image atteignent une valeur désirée en utilisant par exemple une méthode de minimisation de type gradient. Dans ce cas, la trajectoire de la caméra n'est pas du tout contrôlée. Cela peut parfois entraîner des mouvements de la caméra inadéquats (très sous-optimaux), voire impossibles à réaliser. Un exemple maintenant classique, qui illustre très bien ce phénomène, est présenté dans [Chaumette 98].

Considérons un déplacement, entre les images initiale et désirée, composé uniquement d'une rotation de 180 dg autour de l'axe optique de la caméra. Afin de contrôler les mouvements du robot, on choisit les coordonnées des quatre points comme informations visuelles. La trajectoire attendue des points dans l'image est une rotation autour du point principal de la caméra. La trajectoire obtenue pour chaque point d'intérêt est en fait une ligne droite entre ses positions initiale et désirée (figure 1.10), ce qui correspond à un pur mouvement de translation le long de l'axe optique de la caméra l'amenant à l'infini ! Évidemment, on est bien loin d'une trajectoire optimale pour la caméra.

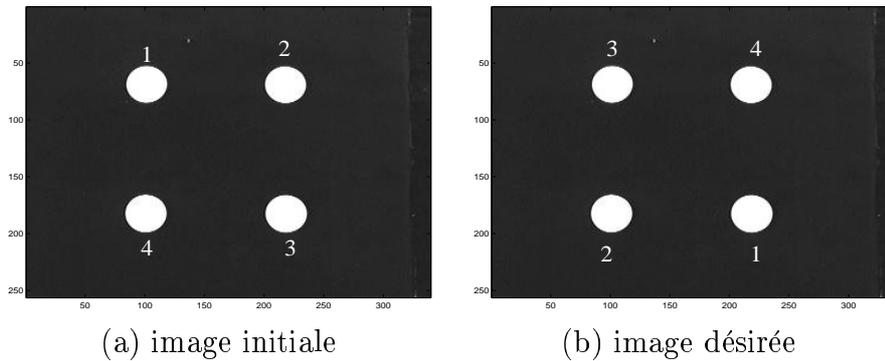


FIG. 1.9 – Images de l'objet d'intérêt

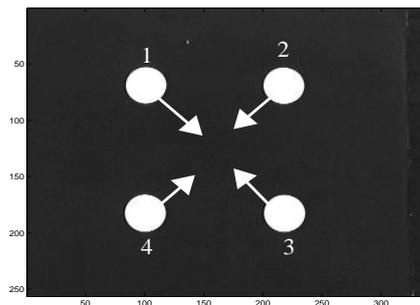


FIG. 1.10 – Trajectoires dans l'image : les points d'intérêt se dirigent en ligne droite vers leurs positions désirées

1.5 Les trajectoires dans l'espace articulaire

En utilisant une consigne fixe, le problème des contraintes mécaniques (butées articulaires) est difficile à gérer dès lors que tous les degrés de liberté du robot sont utilisés pour réaliser la tâche. La trajectoire du robot dans l'espace articulaire peut donc s'avérer inadéquate.

Dans l'exemple suivant, les images correspondant aux positions initiale et désirée sont illustrées sur les Figures 1.11(a) et 1.11(b). Les trajectoires des articulations dans l'espace articulaire, obtenues en utilisant un asservissement 3-D, dans le cas où les modèles sont parfaitement connus, sont données par la Figure 1.12. Les coordonnées dans l'espace articulaire sont normalisées entre $[-1,1]$, où -1 et 1 représentent les butées mécaniques. On observe qu'un des axes atteint sa limite supérieure, ce qui implique une impossibilité à réaliser la tâche. Notons également que ce cas de figure peut se présenter quel que soit le schéma de commande utilisé.

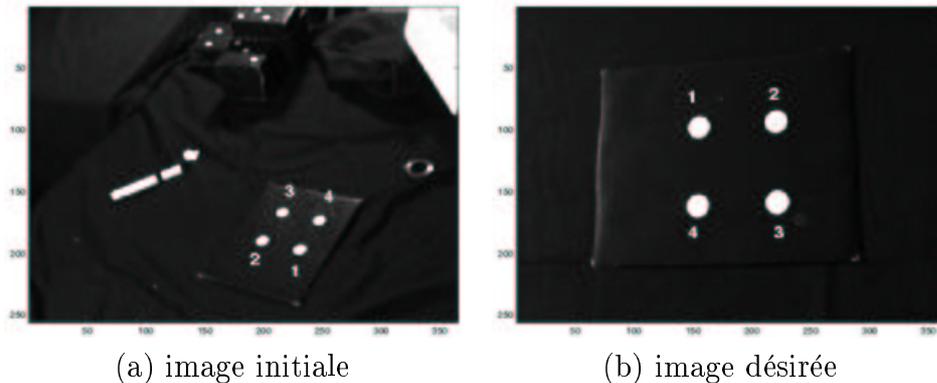


FIG. 1.11 – Images de l'objet d'intérêt

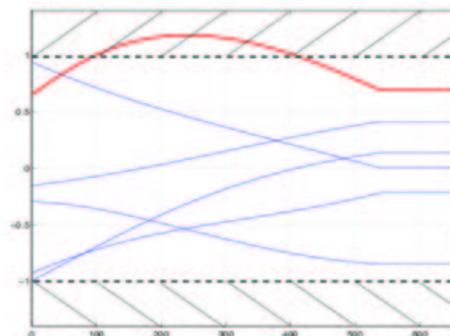


FIG. 1.12 – Trajectoires dans l'espace articulaire: un des axes atteint sa limite supérieure

1.6 Contributions

L'objectif de nos travaux est de répondre aux problèmes que nous avons mis en évidence précédemment. Nos contributions sont les suivantes :

- En ce qui concerne le caractère local des asservissements visuels, nous proposons de coupler la planification de trajectoires à une loi de commande connue comme localement stable et robuste vis-à-vis des erreurs de modélisation. Nous avons développé deux schémas de planification. Nous avons choisi et adapté les asservissements visuels 2D (connus pour être localement stables et robustes) afin de suivre les trajectoires planifiées.
- Pour améliorer significativement les trajectoires du robot dans l'espace de travail, nous avons obtenu les formes analytiques des trajectoires dans l'image correspondant à une trajectoire optimale de l'effecteur du robot.
- Nous avons intégré des contraintes dans la phase de planification des trajectoires dans l'image afin d'assurer en permanence la visibilité de l'objet d'intérêt et afin d'éviter les butées articulaires. Nous avons également traité les problèmes liés aux auto-occultations.
- En outre, les méthodes que nous avons développées ne nécessitent ni les modèles des objets observés ni une calibration précise de la caméra.

Tous les résultats obtenus ont été validés par de nombreuses expérimentations sur la plate-forme robotique de l'IRISA.

1.7 Organisation du manuscrit

Ce mémoire de thèse se divise en deux parties. Dans la première partie, nous rappelons des résultats fondamentaux en géométrie pour la vision et en asservissement visuel. Cette partie est divisée en deux chapitres qui portent, respectivement, sur les points suivants :

- Dans le premier chapitre, nous rappellerons quelques résultats issus de la géométrie épipolaire. Nous discuterons, en particulier, les formulations matricielles de la contrainte épipolaire. Nous présenterons ensuite un bref aperçu des méthodes d'estimation de la géométrie épipolaire et des méthodes de reconstruction du déplacement partiel de la caméra et de la géométrie de la scène, ceci dans les cas où les paramètres de la caméra sont précisément connus ou mal connus.
- Dans le second chapitre, nous dresserons un bref état de l'art des divers travaux relatifs à l'asservissement visuel. Nous décrirons les principales structures de commande permettant la réalisation de tâches robotiques à partir d'informations visuelles extraites d'images fournies par un capteur de vision. Nous rappellerons brièvement l'approche fonction de tâche très largement employée dans ce domaine. Nous présenterons ensuite, de manière plus précise,

les différents schémas de commande 2D, 3D et 2D1/2. Nous nous attacherons à préciser les avantages et les inconvénients de ces trois approches.

La seconde partie est composée de trois chapitres. Le premier chapitre introductif dresse un bref état de l'art du domaine de la planification de trajectoire en robotique. Les deux chapitres suivants décrivent nos contributions. Plus précisément :

- Dans le premier chapitre, nous donnons quelques définitions utiles avant de parcourir l'état de l'art du domaine de la planification de trajectoires en robotique.
- Dans le second chapitre, nous montrerons comment, à partir de deux images d'une scène statique, peut être obtenue une trajectoire continue dans l'image de primitives visuelles correspondant à une trajectoire optimale de la caméra pour aller de la première à la seconde image. Nous résoudrons les problèmes à minimum d'énergie et minimum d'accélération de manière analytique. Nous montrerons que les trajectoires ainsi obtenues sont indépendantes de la calibration de la caméra. Nous utiliserons ensuite un schéma de contrôle de type 2D pour suivre les trajectoires dans l'image. Nous étudierons également le cas de l'interpolation d'un nombre quelconque d'images afin de pallier le problème de mise en correspondance des primitives visuelles lorsque les prises de vue initiale et finale sont distantes. Nous utiliserons ensuite une mise en forme variationnelle du problème de planification de trajectoires dans l'image, afin d'obtenir une solution numérique quand la contrainte de visibilité est prise en compte. Nous présenterons également des résultats expérimentaux validant ces méthodes.
- Nous présenterons, dans le troisième chapitre, une méthode de planification de primitives visuelles dans l'image permettant la prise en compte de contraintes liées au type de capteur utilisé et aux mécanismes du robot. Plus précisément, nous nous intéresserons aux contraintes de visibilité et d'évitement des butées articulaires du robot. Après une description de la méthode classique des fonctions de potentiel, nous verrons comment modifier cette méthode, afin de tenir compte de contraintes relatives à des espaces différents (espace articulaire, espace de travail, espace de tâche). La seconde partie de ce chapitre sera consacrée à l'étude des cas où le modèle de la cible est disponible. Nous étendrons ensuite la méthode proposée au problème de planification quand le modèle de la cible est inconnu. Dans ce cas, nous verrons qu'il est possible d'obtenir des résultats concernant la robustesse du planificateur vis-à-vis d'éventuelles erreurs de modélisation. Dans la quatrième partie, nous utiliserons une interpolation par des fonctions B-spline afin de construire des trajectoires dans l'image continues et dérivables. Nous utiliserons ensuite un schéma d'asservissement visuel 2D pour suivre les trajectoires obtenues. Enfin, la dernière partie du chapitre sera consacrée aux résultats expérimentaux obtenus sur la plate-forme robotique de l'IRISA.

Première partie
Rappels et motivations

Chapitre 2

Géométrie d'un système muni d'une caméra

Le problème de la localisation tridimensionnelle d'une caméra dans la scène qu'elle observe peut classiquement être traité de deux manières différentes selon le degré de connaissance a priori sur la caméra et sur la scène observée. Le premier type de méthodes utilise une mise en correspondance entre des points de l'espace Cartésien et leur projection dans l'image de la caméra pour obtenir la **pose** de l'objet d'intérêt dans un repère rigidement lié à la caméra. Dans la suite, on désignera par pose un élément de $SE(3)$ (Special Euclidean Group), c'est-à-dire la caractérisation de la situation d'un solide par sa position et son orientation dans un repère choisi. Ce type de méthode implique que l'on dispose d'un modèle de la scène ou tout du moins d'un certain nombre de primitives de celle-ci [Dementhon 95, Lowe 91, Dhome 89]. Le déplacement de la caméra peut ensuite être obtenu par composition de changements de repère entre deux poses successives.

Le principe du second groupe de méthodes, que nous décrirons plus en détail dans la suite de ce chapitre, est de retrouver, à partir de la mise en correspondance de primitives géométriques dans plusieurs images, la géométrie de la scène observée. Dans ce cas, le modèle de la scène n'est pas nécessaire. Si la calibration du capteur est connue, on parlera de reconstruction euclidienne; sinon la reconstruction sera dite projective. Dans le cas d'une reconstruction euclidienne, le déplacement en translation de la caméra entre les prises de vue ne pourra être obtenu qu'à un facteur d'échelle près; on parlera alors de **reconstruction partielle du déplacement** de la caméra.

Dans la suite de ce chapitre, nous rappelons tout d'abord quelques résultats de géométrie projective permettant de caractériser de manière simple les transformations rigides de l'espace Cartésien et les transformations projectives réalisées par la caméra. Nous rappelons également quelques résultats issus de la géométrie épipolaire. En particulier, nous donnons les formulations matricielles de la contrainte épipolaire. Nous présentons ensuite un bref aperçu des méthodes d'estimation de la géométrie épipolaire et de reconstruction partielle du déplacement de la caméra et de

la géométrie de la scène, dans les cas où les paramètres de la caméra sont précisément ou mal connus. Nous aurons recours à de nombreuses reprises à ces résultats dans toute la suite du document.

2.1 Espace projectif et transformation homogène

La géométrie projective constitue un outil puissant de modélisation en vision par ordinateur et est de ce fait très utilisée dans ce domaine. Elle permet une résolution élégante de problèmes primordiaux en vision, citons par exemple :

- l'estimation du déplacement partiel de la caméra entre deux prises de vue et la reconstruction de scène à partir de plusieurs images [Faugeras 87], [Faugeras 88], [Hartley 92], [Huang 89],[Viéville 96],
- l'estimation des paramètres de la caméra [Zhang 96c],
- l'indexation d'images par l'utilisation d'invariants projectifs [Schmid 97],
- le positionnement d'un robot relativement à une scène inconnue [Malis 99].

La caméra est un capteur qui réalise une projection d'un ensemble de points d'une scène tridimensionnelle dans un espace à deux dimensions : le plan image. Le cadre naturel pour étudier de manière simple de telles transformations est la géométrie projective. Nous allons voir qu'il est également intéressant d'englober l'espace cartésien de dimension n , \mathbb{R}^n , dans l'espace projectif \mathcal{P}^n afin de manipuler simplement les transformations affines agissant sur \mathbb{R}^n . En effet, cette opération va permettre de formuler linéairement ce type de transformation mais aussi de manipuler des entités géométriques à l'infini. Soit ${}^x\bar{\mathbf{M}} = [X_1 \ X_2 \ \cdots \ X_n]^T \in \mathbb{R}^n$ le vecteur des coordonnées cartésiennes d'un point \mathcal{M} de l'espace euclidien de dimension n . Ce même point peut être décrit, à un facteur d'échelle près, dans l'espace projectif \mathcal{P}^n par ${}^x\mathbf{M} = [\lambda X_1 \ \lambda X_2 \ \cdots \ \lambda X_n \ \lambda]^T \in \mathbb{R}^{n+1}$. Si λ est non nul alors on retrouve simplement les coordonnées du point dans l'espace euclidien en divisant toutes les coordonnées par la dernière coordonnée projective. Si λ est nul alors ${}^x\mathbf{M}$ représente une direction de l'espace euclidien, ce qui correspond à un point \mathcal{M} à l'infini. On notera T_x la dernière coordonnée de ${}^x\mathbf{M}$. Plaçons-nous maintenant dans l'espace Euclidien de dimension 3. Soit \mathcal{F}_x et \mathcal{F}_y deux repères de cet espace. La matrice de transformation homogène entre \mathcal{F}_x et \mathcal{F}_y s'écrit :

$${}^xT_y = \begin{bmatrix} {}^x\mathbf{R}_y & {}^x\mathbf{t}_y \\ 0 & 1 \end{bmatrix} \quad (2.1)$$

où ${}^x\mathbf{R}_y$ et ${}^x\mathbf{t}_y$ sont respectivement la matrice de rotation et le vecteur de translation entre \mathcal{F}_x et \mathcal{F}_y , les indices x à gauche et y à droite signifiant que le repère \mathcal{F}_y est exprimé dans le repère \mathcal{F}_x . La transformation inverse homogène entre \mathcal{F}_y et \mathcal{F}_x peut alors s'écrire :

$${}^xT_y^{-1} = \begin{bmatrix} {}^x\mathbf{R}_y^T & -{}^x\mathbf{R}_y^T {}^x\mathbf{t}_y \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^y\mathbf{R}_x & {}^y\mathbf{t}_x \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

Ces matrices de transformation homogènes permettent d'exprimer simplement les changements de repère. Ainsi un point \mathcal{M} de coordonnées homogènes ${}^x\mathbf{M}$ dans \mathcal{F}_x aura pour coordonnées homogènes ${}^y\mathbf{M}$, dans \mathcal{F}_y , avec:

$${}^y\mathbf{M} = {}^y\mathbf{T}_x {}^x\mathbf{M} \quad (2.3)$$

Un plan Π représenté dans \mathcal{F}_x par le vecteur ${}^x\boldsymbol{\pi} = [\mathbf{n}^{xT} \ d^x]^T$, où \mathbf{n}^x désigne son vecteur normal exprimé dans \mathcal{F}_x et d^x la distance de Π à l'origine C_x de \mathcal{F}_x , est caractérisé dans le repère \mathcal{F}_y par le vecteur ${}^y\boldsymbol{\pi} = [\mathbf{n}^{yT} \ d^y]^T$ avec :

$${}^y\boldsymbol{\pi} = {}^y\mathbf{T}_x^{-T} {}^x\boldsymbol{\pi} \quad (2.4)$$

L'utilisation des matrices homogènes permet donc d'obtenir une formulation linéaire des opérations de changement de repère. Dans la suite de l'étude, on supposera que les robots, les caméras et les objets d'intérêt évoluent dans un espace \mathcal{W} , que l'on appellera espace de travail et qui constitue un sous-ensemble de l'espace cartésien \mathbb{R}^3 . Nous utiliserons dans la suite du document un certain nombre de repères qui nous permettront de représenter la situation d'un solide dans l'espace \mathcal{W} (voir Figure 2.1):

- $\mathcal{F}_c(C, X_c, Y_c, Z_c)$ représente le repère lié à la caméra dans sa position courante où :
 - C est le centre optique de la caméra,
 - Z_c est confondu avec l'axe optique de la caméra,
 - X_c et Y_c sont, respectivement, parallèles aux lignes et aux colonnes de l'image.
- $\mathcal{F}_i(C_i, X_i, Y_i, Z_i)$ représente le repère lié à la caméra lorsque celle-ci est dans sa position initiale
- $\mathcal{F}_f(C_f, X_f, Y_f, Z_f)$ est le repère lié à la caméra lorsque celle-ci est dans sa position désirée
- $\mathcal{F}_o(O, X_o, Y_o, Z_o)$ est le repère dans lequel est défini le modèle géométrique de l'objet observé par la caméra, lorsque ce modèle est disponible.

La situation d'un solide, auquel on a attaché un repère \mathcal{F}_y peut être représentée dans un repère \mathcal{F}_x par la matrice homogène ${}^x\mathbf{T}_y$. Cependant, une telle représentation est redondante car seulement trois variables suffisent pour identifier une matrice de rotation dans le groupe $SO(3)$ des rotations. Nous choisirons plutôt la représentation des rotations par le vecteur de dimension 3, $\boldsymbol{\theta} = \mathbf{u}\theta$, où \mathbf{u} représente le vecteur unitaire porté par l'axe de la rotation et θ représente l'angle de la rotation. D'une part, cette représentation de la rotation a l'avantage d'être minimale et peut facilement être obtenue à partir de la matrice de rotation associée. D'autre part, le jacobien liant les variations temporelles de cette paramétrisation et le torseur cinématique de la caméra présente une singularité uniquement pour $\theta = 2k\pi$ (avec $k \in \mathbb{Z}^*$), situées en dehors de l'espace de travail. Remarquons, par exemple que les représentations $\mathbf{u} \sin(\theta)$ et $\mathbf{u} \sin(\frac{\theta}{2})$ présentent respectivement une singularité de leur jacobien en $\frac{\pi}{2}$ et π .

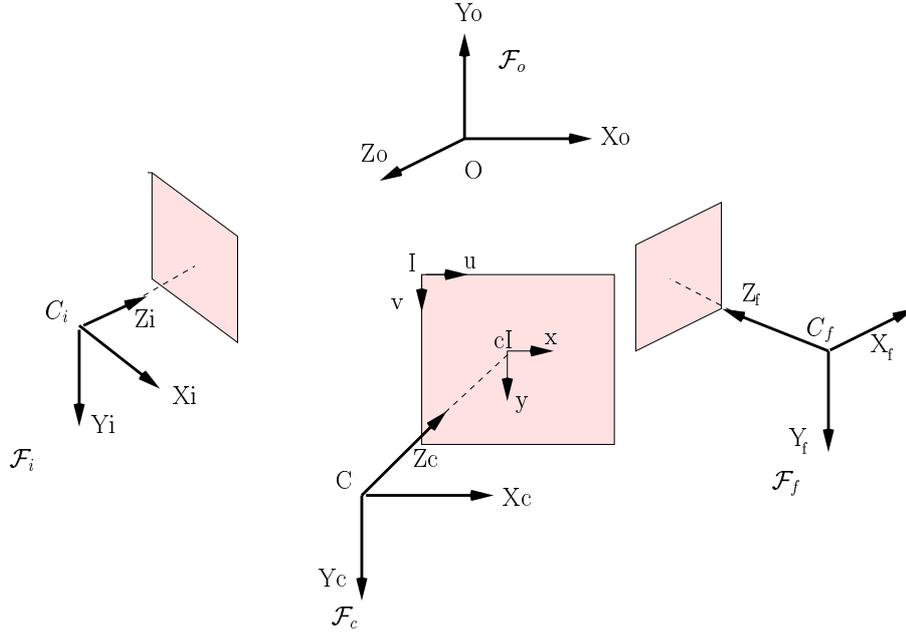


FIG. 2.1 – Les repères

2.2 Modélisation de la caméra

Le capteur caméra est décrit par le modèle projectif dit sténopé. Dans ce modèle, la caméra réalise une projection perspective de centre C (origine du repère \mathcal{F}_c et centre optique de la caméra) des points de \mathcal{P}^3 sur le plan image \mathcal{I} , situé à une distance f du centre optique, en des points de \mathcal{P}^2 (se référer à la figure 2.2). Cette transformation peut être décomposée en trois étapes :

- étape 1 : soit \mathcal{M} un point de \mathcal{P}^3 de coordonnées homogènes $\mathbf{M}_x = [X_x \ Y_x \ Z_x \ T_x]^T$ dans un repère \mathcal{F}_x (en général on choisit $\mathcal{F}_x = \mathcal{F}_o$), alors ce point est représenté par \mathbf{M}_c dans le repère \mathcal{F}_c , avec :

$${}^c\mathbf{M} = [X_c \ Y_c \ Z_c \ T_c]^T = {}^c\mathbf{T}_x {}^x\mathbf{M} \quad (2.5)$$

- étape 2 : le point \mathcal{M} exprimé dans le repère de la caméra \mathcal{F}_c est projeté en un point de l'espace image \mathcal{I} de coordonnées projectives $\mathbf{m} = [fm_x \ fm_y \ f]^T$ avec :

$$\frac{Z_c}{f}\mathbf{m} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^c\mathbf{T}_x {}^x\mathbf{M} = \underbrace{{}^c\mathbf{P} {}^c\mathbf{T}_x}_{{}^x\mathbf{P}} {}^x\mathbf{M} \quad (2.6)$$

où ${}^c\mathbf{P}$ et ${}^x\mathbf{P} = [{}^c\mathbf{R}_x \ {}^c\mathbf{t}_x]$ sont respectivement les matrices de projection exprimées dans \mathcal{F}_c et \mathcal{F}_x . En outre, on peut fixer $f = 1$ car les coordonnées d'un point dans un espace projectif sont définies à un facteur d'échelle près.

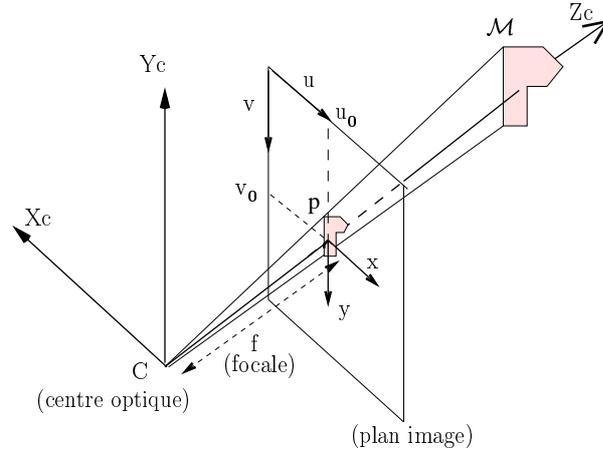


FIG. 2.2 – Modélisation de la caméra

- étape 3 : La caméra effectue également une transformation des coordonnées homogènes métriques $\mathbf{m} = [m_x \ m_y \ 1]^T$ en coordonnées homogènes exprimées en pixel $\mathbf{p} = [u \ v \ 1]^T$:

$$\mathbf{p} = \mathbf{K}\mathbf{m} \quad (2.7)$$

où \mathbf{K} , la matrice des paramètres intrinsèques de la caméra est une matrice 3×3 dont les coefficients peuvent être déterminés si besoin par une procédure de calibration [Faugeras 86], [Tsai 86]. Elle s'écrit :

$$\mathbf{K} = \begin{bmatrix} -fk_u & fk_u \cot \phi & u_0 \\ 0 & -\frac{fk_v}{\sin \phi} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & \alpha_{uv} & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

- k_u et k_v sont les facteurs d'échelle horizontal et vertical des pixels,
- u_0 et v_0 sont les coordonnées du point principal défini comme étant l'intersection entre l'axe optique et le plan image.
- ϕ est l'angle entre les axes du repère image.

On obtient la transformation réalisée par la caméra d'un point de \mathcal{P}^3 , dont les coordonnées sont exprimées dans un repère \mathcal{F}_x , en un point du plan image \mathcal{I} , dont les coordonnées sont exprimées en pixel, à partir des équations (2.6) et (2.7) :

$$\frac{Z_c}{f} \mathbf{p} = \mathbf{K} {}^x\mathbf{P} {}^x\mathbf{M} \quad (2.9)$$

Il existe dans la littérature d'autres modèles de caméra (projection orthographique, perspective faible, para perspective, ...). Le lecteur intéressé pourra par exemple se référer à [Boufama 94], [Horaud 93], [Faugeras 93]. Notons également que le modèle projectif de la caméra que nous utiliserons est linéaire car la caméra est supposée parfaite (i.e sans distorsion). Pour des modèles de caméra avec distorsion, on pourra consulter [Li 95, Zhang 96b, Rémy 98].

2.3 Reconstruction projective et euclidienne

La géométrie d'un système à deux prises de vue a donné lieu à une importante littérature. Il existe, en effet, une contrainte forte, liant les deux projections d'un point de l'espace dans les deux images, qui conduit à des résultats importants. Considérons un point \mathbf{p}_1 appartenant à la première image \mathcal{I}_1 (se référer à la Figure 2.3). Le point \mathcal{M} de l'espace qui s'y projette appartient à la droite $(C_1\mathbf{p}_1)$. Le point correspondant \mathbf{p}_2 dans la deuxième image \mathcal{I}_2 doit donc appartenir à la projection de la droite $(C_1\mathbf{p}_1)$ dans la seconde image. Cette droite projetée est communément appelée droite épipolaire de \mathcal{I}_2 associée à \mathbf{p}_1 . L'ensemble des droites épipolaires de \mathcal{I}_2 s'intersecte alors en un même point qui est la projection de C_1 dans \mathcal{I}_2 . Ce point particulier s'appelle l'épipole et est noté \mathbf{e}_2 . De la même manière, l'ensemble des droites épipolaires de la première image s'intersecte en l'épipole de \mathcal{I}_1 , \mathbf{e}_1 , qui est la projection de C_2 dans \mathcal{I}_1 . Cette contrainte, appelée contrainte épipolaire est, en général, exploitée sous ses formes matricielles. Dans le prochain paragraphe, nous explicitons ces relations matricielles. Ensuite, nous présentons brièvement les techniques d'estimation de la géométrie épipolaire et du déplacement partiel de la caméra entre les deux prises de vue.

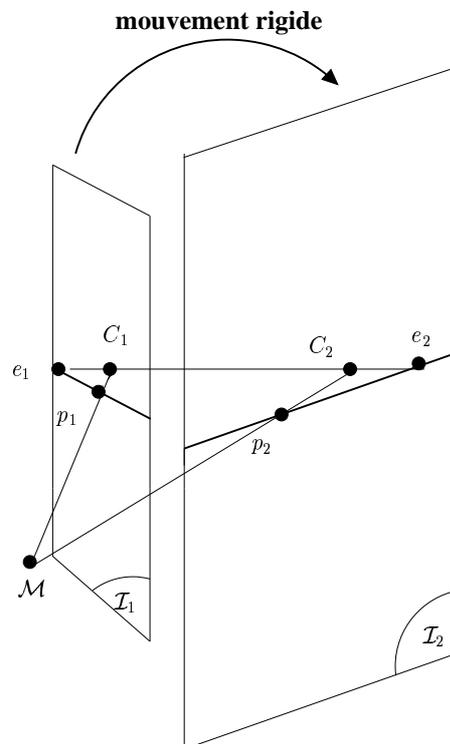


FIG. 2.3 – La géométrie épipolaire

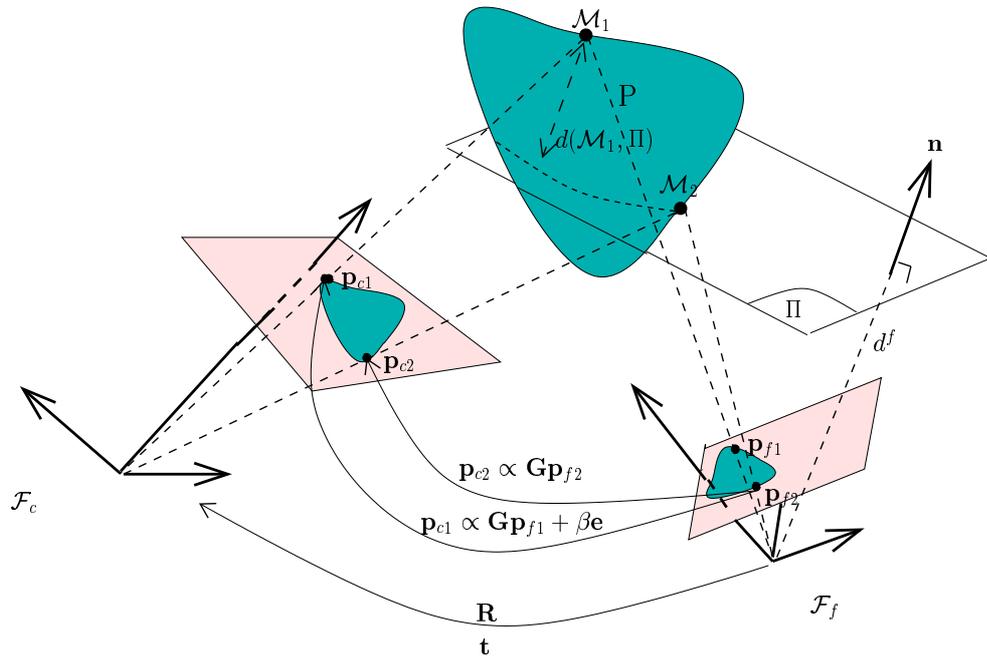


FIG. 2.4 – Les contraintes entre deux images

2.3.1 Les expressions matricielles de la contrainte épipolaire

Le problème classique de reconstruction du déplacement d'une caméra entre deux prises de vue, lorsqu'on ne dispose pas d'informations précises sur la géométrie de la scène est, en général, décomposé en deux étapes. La première consiste en l'estimation de la géométrie épipolaire à partir de la mise en correspondance d'un certain nombre de primitives géométriques dans les deux images. La seconde étape consiste ensuite à utiliser les relations liant les expressions matricielles de la contrainte épipolaire pour construire des méthodes permettant de retrouver le déplacement partiel de la caméra et la structure de la scène. Dans ce paragraphe, nous allons brièvement décrire les relations matricielles qui nous intéresseront dans la suite du document. Pour une description plus complète, le lecteur pourra se reporter à [Faugeras 93]. Dans le contexte de l'asservissement visuel, l'objectif est d'établir une loi de commande afin de faire converger une image, acquise lorsque la caméra est dans la position définie par le repère \mathcal{F}_c , vers une image de référence, acquise lorsque la caméra est dans sa position désirée \mathcal{F}_f . Les projections perspectives d'une scène statique, lorsque la caméra effectue un mouvement l'amenant de sa position courante à sa position désirée, sont liées par un ensemble de relations que nous allons expliciter (cf Figure 2.4).

Choisissons un plan Π de référence attaché à la scène que nous caractérisons dans le repère \mathcal{F}_f par le vecteur de dimension 4, $\boldsymbol{\pi}^f = [\mathbf{n}^f \quad d^f]^T$ composé des coordonnées \mathbf{n}^f du vecteur normale unitaire à Π , et de la distance d^f du centre de projection de la caméra dans sa position désirée à Π . Le plan Π sera caractérisé de manière

similaire dans \mathcal{F}_c par le vecteur $\boldsymbol{\pi} = [\mathbf{n}^T \quad d]^T$. On définit la distance signée d'un point 3D \mathcal{M} de la scène, de coordonnées homogènes $\mathbf{M}_f = [X_f \quad Y_f \quad Z_f \quad T_f]^T$ dans \mathcal{F}_f et $\mathbf{M}_c = [X_c \quad Y_c \quad Z_c \quad T_c]$ dans \mathcal{F}_c , au plan Π par :

$$d(\mathcal{M}, \Pi) = \boldsymbol{\pi}^{fT} \mathbf{M}_f \quad (2.10)$$

Le point \mathcal{M} se projette dans l'image courante et désirée en deux points de coordonnées homogènes exprimées en mètre \mathbf{m} et \mathbf{m}_f , respectivement, et ces coordonnées sont liées par la relation suivante :

$$Z_c \mathbf{m} = Z_f \mathbf{H} \mathbf{m}_f + \varepsilon {}^c \mathbf{t}_f \quad (2.11)$$

où \mathbf{H} est une matrice 3×3 non singulière qui est dépendante du choix du plan de référence Π . Elle réalise une transformation linéaire de \mathcal{P}^2 dans \mathcal{P}^2 (*i.e* du plan image dans le plan image) pour tous les points issus de la projection du plan de référence. Pour tous ces points, la relation (2.11) se simplifie en :

$$Z_c \mathbf{m} = Z_f \mathbf{H} \mathbf{m}_f \quad (2.12)$$

La matrice \mathbf{H} est communément appelée **matrice d'homographie** relative au plan Π et joue un rôle primordial dans l'estimation du déplacement de la caméra et de la géométrie de la scène. Elle peut en effet être décomposée en la somme de la matrice de rotation entre \mathcal{F}_c et \mathcal{F}_f et d'une matrice de rang un où apparaît le vecteur normal à Π et le vecteur de translation entre \mathcal{F}_c et \mathcal{F}_f (pour simplifier l'exposé, on notera dans la suite \mathbf{R} et \mathbf{t} la matrice de rotation ${}^c \mathbf{R}_f$ et le vecteur de translation ${}^c \mathbf{t}_f$) :

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{t}}{d^f} \mathbf{n}^{fT} \quad (2.13)$$

Par ailleurs, le scalaire ε , apparaissant dans la relation (2.11), est donné par la relation suivante :

$$\varepsilon = - \frac{d(\mathcal{M}, \Pi)}{d^f} \quad (2.14)$$

Rappelons que ε est nul si le point \mathcal{M} appartient au plan de référence Π . On retrouve donc dans ce cas la relation (2.12) à partir de la relation (2.11). Notons également le rôle particulier joué par l'homographie \mathbf{H}_∞ relative au plan situé à l'infini Π_∞ dans l'estimation du mouvement et de la structure. En effet, son utilisation permet de simplifier de manière significative ce processus. Si on choisit le plan à l'infini Π_∞ comme plan de référence alors $\mathbf{n}^{fT}/d^f = 0$ et $\varepsilon = 1$, l'équation (2.11) s'écrit alors :

$$Z_c \mathbf{m} = Z_f \mathbf{H}_\infty \mathbf{m}_f + \mathbf{t} \quad (2.15)$$

et la relation (2.13) se simplifie en :

$$\mathbf{H}_\infty = \mathbf{R} \quad (2.16)$$

La matrice d'homographie pour un plan de référence Π quelconque s'écrit en fonction de \mathbf{H}_∞ de la manière suivante :

$$\mathbf{H} = \mathbf{H}_\infty + \frac{\mathbf{t}}{df} \mathbf{n}^{fT} \quad (2.17)$$

On obtient aisément une relation similaire à la relation (2.11) dans le système de coordonnées en pixel en utilisant l'équation (2.7) :

$$Z_c \mathbf{p}_c = Z_f \mathbf{K} \mathbf{H} \mathbf{K}^{-1} \mathbf{p}_f + \varepsilon \mathbf{K} \mathbf{t} \quad (2.18)$$

La nature de cette dernière relation n'est pas changée si on multiplie chacun de ces membres par un scalaire α non nul. En tenant compte de la remarque précédente, on écrira dans la suite l'équation (2.18) sous la forme plus compacte :

$$\lambda \mathbf{p} = \mathbf{G} \mathbf{p}_f + \beta \mathbf{e} \quad (2.19)$$

où :

- la matrice \mathbf{G} est une matrice non singulière de dimension 3. Elle dépend du choix du plan de référence et est exprimée dans le système de coordonnées pixéliques. Plus précisément on a :

$$\mathbf{G} = \alpha \mathbf{K} \mathbf{H} \mathbf{K}^{-1} \quad (2.20)$$

Elle est, en général, également appelée matrice d'homographie. Dans la suite, on la nommera plutôt **matrice de colinéation** afin de la différencier de la matrice \mathbf{H} ;

- le vecteur \mathbf{e} est la projection dans l'image courante du centre du repère \mathcal{F}_f , c'est-à-dire l'épipole dans l'image courante :

$$\mathbf{e} = \mathbf{K} \mathbf{t} \quad (2.21)$$

- λ est un scalaire non nul avec :

$$\lambda = \alpha Z_c / Z_f \quad (2.22)$$

- β est un scalaire nul quand le point \mathcal{M} appartient au plan de référence Π et est donné dans le cas général par :

$$\beta = \alpha \varepsilon / Z_f \quad (2.23)$$

Les équations de projection du point \mathcal{M} dans les deux images sont données par l'équation (2.9) :

$$\begin{aligned} \frac{Z_f}{f} \mathbf{p}_f &= \mathbf{K}^f \mathbf{P}_o \mathbf{M}_o \\ \frac{Z_c}{f} \mathbf{p} &= \mathbf{K}^c \mathbf{P}_o \mathbf{M}_o \end{aligned} \quad (2.24)$$

avec :

$${}^c\mathbf{P}_o = [{}^c\mathbf{R}_o \quad {}^c\mathbf{t}_o] \quad \text{et} \quad {}^f\mathbf{P}_o = [{}^f\mathbf{R}_o \quad {}^f\mathbf{t}_o] \quad (2.25)$$

En éliminant \mathbf{M}_o dans les deux équations précédentes, on obtient la relation suivante entre les vecteurs \mathbf{p} et \mathbf{p}_f :

$$\mathbf{p}^T \mathbf{F} \mathbf{p}_f = 0 \quad (2.26)$$

avec :

$$\mathbf{F} = \mathbf{K}^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K} = [\mathbf{e}]_{\times} \mathbf{G} \quad (2.27)$$

où $[\mathbf{x}]_{\times}$ représente la matrice de pré-produit vectoriel associée au vecteur \mathbf{x} . \mathbf{F} est une matrice de dimension 3×3 de rang au plus 2 appelée **matrice fondamentale**. La littérature concernant les propriétés, l'estimation et l'utilisation de la matrice fondamentale dans le domaine de la vision par ordinateur est très importante. On pourra par exemple se reporter à [Luong 96, Faugeras 93, Horaud 93] pour plus de détails. Dans le système des coordonnées images métriques, la **matrice essentielle** \mathbf{E} introduite par H.C Longuet-Higgins, joue un rôle similaire à la matrice fondamentale [Luong 96, LonguetHiggins 81, Hartley 97]. Elle vérifie la relation :

$$\mathbf{m}^T \mathbf{E} \mathbf{m}_f = 0 \quad (2.28)$$

et, d'après l'équation (2.27), on a :

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (2.29)$$

Les différentes expressions matricielles exprimant la contrainte épipolaire liant les projections dans deux images d'un même point 3D sont très importantes et permettent une résolution élégante de problèmes difficiles dans le domaine de la vision. Nous allons, dans le paragraphe suivant, donner un bref aperçu des méthodes d'estimation de la géométrie épipolaire (c'est-à-dire des matrices \mathbf{G} , \mathbf{H} , \mathbf{F} , \mathbf{E} et du vecteur \mathbf{e}) et de son application dans l'estimation du déplacement partiel de la caméra et de la géométrie de la scène.

2.3.2 Estimation de la géométrie épipolaire et du déplacement euclidien

Nous considérerons uniquement les cas où la caméra fournit une image sans distorsion optique ou géométrique et est décrite par le modèle sténopé. On pourra se reporter à [Zhang 96b] lorsque ces hypothèses ne sont pas valides. Nous ne présentons ici qu'un bref aperçu des méthodes d'estimation de la géométrie épipolaire et invitons les lecteurs désirant plus de précisions à se reporter à [Luong 93, Torr 97, Zhang 96a].

A partir de la mise en correspondance d'un ensemble de n points dans les deux images, il est possible de calculer la matrice fondamentale \mathbf{F} et/ou la colinéation \mathbf{G}

relative à un plan de référence Π . Cela nécessite, dans le cas général, un nombre de points $n \geq 8$, et un nombre de points $n \geq 4$ dans le cas particulier où les points considérés sont coplanaires.

L'estimation de la géométrie épipolaire est un problème, par nature, non linéaire. La matrice fondamentale doit être de rang 2 et la matrice essentielle doit satisfaire les conditions de Huang-Faugeras [Huang 89]. L'approche classique pour estimer la géométrie épipolaire est en général composée de deux étapes. Dans un premier temps, un algorithme linéaire fournit une solution. Cette solution est ensuite affinée en introduisant les contraintes non-linéaires (par exemple imposer un rang de 2 à la matrice fondamentale) via un algorithme de minimisation [Jerian 91]. La solution obtenue en utilisant l'algorithme linéaire des huit points proposé dans [LonguetHiggins 81, LonguetHiggins 84, Hartley 97] peut, par exemple, être utilisée pour initialiser les algorithmes non linéaires décrits dans [Deriche 94, Luong 96]. Afin de simplifier l'estimation de la matrice fondamentale, celle-ci peut être décomposée en utilisant la relation (2.27). L'estimation de la matrice fondamentale revient alors à estimer l'épipole et la matrice de colinéation. En effectuant un changement de base judicieux, Boufama et *al.* ont proposé d'estimer simultanément la matrice de colinéation et l'épipole, ce qui permet d'obtenir directement une matrice fondamentale de rang 2 [Boufama 95]. Sashua et *al.* proposent d'estimer dans un premier temps l'épipole et ensuite la matrice d'homographie [Shashua 94]. Si on cherche à estimer uniquement \mathbf{G} , on pourra par exemple utiliser les algorithmes décrits dans [Couapel 95, Malis 00]. Lorsqu'on dispose de quatre points coplanaires, le problème est plus simple puisqu'il se résume à la résolution d'un système linéaire [Criminisi 98]. Nous utiliserons dans nos expérimentations l'algorithme décrit dans [Malis 00].

Certaines configurations entraînent une dégénérescence (réduction de rang) de la matrice fondamentale. Ces configurations sont décrites dans [Torr 98]; parmi les plus fréquentes, notons les trois suivantes :

- lorsque le déplacement de la caméra est une rotation pure,
- lorsque la scène est composée d'un plan unique,
- lorsque tous les points 3D se projettent sur une droite unique.

Dans le contexte de l'asservissement visuel, ces configurations ne sont pas rares. Notons en particulier qu'au voisinage de la position désirée, le déplacement à réaliser doit être nul. Nous nous trouvons alors dans la configuration du premier point cité précédemment. Nous préférons donc l'utilisation de la colinéation à celle de la matrice fondamentale, la colinéation relative à un plan ayant l'avantage de ne jamais être dégénérée. La matrice \mathbf{G} peut être estimée à partir d'un ensemble de droites [Faugeras 87] mises en correspondance dans les deux images, ou encore en utilisant des contours modélisés par des fonctions polynômiales [Chesi 00].

Pour estimer la colinéation relative au plan à l'infini, il faut disposer d'au moins quatre images de la scène ou de connaissances supplémentaires sur la scène, telles que trois points situés à l'infini, ou encore utiliser un plan de référence parallèle au plan image. Pour plus de précisions sur l'estimation de cette colinéation particulière, on

pourra se reporter à [Robert 95, Viéville 96]. Lorsque le mouvement de la caméra est une pure rotation où une pure translation alors la matrice de colinéation à l'infini \mathbf{G}_∞ est facilement obtenue. En effet, dans le premier cas elle correspond à la matrice de colinéation \mathbf{G} relative à un plan quelconque de la scène et dans le second cas elle est proportionnelle à la matrice identité ($\mathbf{G}_\infty \propto \mathbf{I}$).

Une fois la géométrie épipolaire obtenue, il est possible de remonter à la géométrie euclidienne. C'est le problème bien connu de “*structure from motion*”. On pourra donc obtenir une reconstruction géométrique de la scène observée par la caméra (“*structure*”), et le déplacement partiel de la caméra entre deux prises de vue (“*motion*”). Plus précisément, connaissant la géométrie épipolaire entre deux images I_c et I_f (c'est-à-dire la matrice fondamentale \mathbf{F} ou la matrice de colinéation \mathbf{G} relative à un plan de référence), on peut estimer la rotation de la caméra et sa translation à un facteur d'échelle près. La première étape consiste en l'estimation des matrices essentielle et/ou d'homographie à partir des matrices fondamentale et/ou de colinéation. Ces matrices peuvent être estimées de la manière suivante :

$$\hat{\mathbf{E}} = \mathbf{K}^T \mathbf{F} \mathbf{K} \quad (2.30)$$

$$\hat{\mathbf{H}} = \mathbf{K}^{-1} \mathbf{G} \mathbf{K} \quad (2.31)$$

Une connaissance des paramètres intrinsèques de la caméra est donc nécessaire. Lorsque les paramètres intrinsèques sont entachés d'erreurs, l'estimation du déplacement de la caméra et de la structure de la scène sera également entachée d'erreurs. Cependant, nous verrons dans la suite que connaissant la matrice de colinéation à l'infini certains paramètres de mouvement et de structure peuvent être exprimés en fonction des valeurs réelles et des erreurs de calibration. Comme nous l'avons déjà vu (eq. (2.13)), la matrice d'homographie réelle peut s'écrire comme la somme d'une matrice de rotation et d'une matrice de rang 1, ce qui se traduit par la relation suivante :

$$\hat{\mathbf{H}} = \alpha \mathbf{H} = \alpha \left(\mathbf{R} + \frac{\mathbf{t}}{d^f} \mathbf{n}^f \mathbf{n}^{fT} \right) \quad (2.32)$$

A partir de la matrice $\hat{\mathbf{E}}$, on peut estimer la rotation \mathbf{R} , la translation à un facteur d'échelle près, $\frac{\mathbf{t}}{d^f}$ et la normale \mathbf{n}^f au plan Π en utilisant la méthode décrite par Hartley dans [Hartley 92] basée sur la décomposition SVD de la matrice $\hat{\mathbf{E}}$. Afin d'estimer ces paramètres, on peut également utiliser les algorithmes proposés dans [Faugeras 88] ou [Zhang 95b] basés sur la décomposition (2.32) de la matrice $\hat{\mathbf{H}}$. Dans ce dernier cas de figure, on obtient, en général, deux solutions différentes. Cette indétermination peut cependant être levée :

- dans le cas d'une cible plane, si l'on dispose d'informations géométriques additionnelles, par exemple la connaissance approximative de la normale au plan de référence,

- dans le cas d'une cible non plane en considérant un second plan de référence et en choisissant la solution commune [Faugeras 88].

Après avoir estimé le déplacement partiel de la caméra, il est possible de remonter à des informations sur la géométrie de la scène. On peut, par exemple, obtenir les paramètres suivants que nous utiliserons à de nombreuses reprises dans la suite du document :

- le ratio r entre les distances d et d^f des centres de projection C et C_f au plan de référence Π :

$$r = \frac{d}{d^f} = \det(\mathbf{H}) = 1 + \mathbf{n}^{fT} \mathbf{R} \frac{\mathbf{t}}{d^f} \quad (2.33)$$

- le rapport τ entre les profondeurs courante Z_c et désirée Z_f d'un point 3D \mathcal{M} :

$$\begin{cases} \tau = \frac{Z_c}{Z_f} = \frac{\mathbf{n}^{fT} \mathbf{m}_f}{\mathbf{n}^T \mathbf{m}} & \text{si } \mathcal{M} \in \Pi \\ \tau = \frac{Z_c}{Z_f} = \frac{\|[\mathbf{t}]_{\times} \mathbf{R} \mathbf{m}_f\|}{\|[\mathbf{t}]_{\times} \mathbf{m}\|} & \text{si } \mathcal{M} \notin \Pi \end{cases} \quad (2.34)$$

- le rapport ρ entre la profondeur courante Z_c et la distance d^f :

$$\begin{cases} \rho = \frac{Z_c}{d^f} = \frac{r}{\mathbf{n}^T \mathbf{m}} & \text{si } \mathcal{M} \in \Pi \\ \rho = \frac{Z_c}{d^f} = \tau \frac{\|\mathbf{t}/d^f\|}{\|\mathbf{t}/Z_f\|} & \text{si } \mathcal{M} \notin \Pi \end{cases} \quad (2.35)$$

avec :

$$\frac{\mathbf{t}}{Z_f} = \tau \mathbf{m} - \mathbf{R} \mathbf{m} \quad (2.36)$$

Dans [Malis 00], une étude comparative entre différentes méthodes de reconstruction des paramètres de mouvement a conclu sur la plus grande robustesse des méthodes utilisant une estimation de la matrice d'homographie. Cette matrice et les paramètres qu'elle permet de reconstruire seront utilisés dans la suite pour la planification de trajectoire lorsqu'on ne dispose pas de modèle de la cible. En outre, quand la calibration de la caméra est mal connue, il est possible d'obtenir l'expression analytique des paramètres estimés en fonction des paramètres réels et des erreurs de calibration [Malis 99]. Ceci nous permettra d'étudier la sensibilité de nos méthodes de planification de trajectoires vis-à-vis d'éventuelles erreurs de modélisation.

Considérons la matrice contenant les paramètres intrinsèques de la caméra comme erronée et donnée par $\widehat{\mathbf{K}}$, et supposons que la matrice de colinéation à l'infini \mathbf{G}_{∞} puisse être estimée. Alors la matrice d'homographie à l'infini peut être estimée de la manière suivante :

$$\widehat{\mathbf{H}}_{\infty} = \widehat{\mathbf{K}}^{-1} \mathbf{G}_{\infty} \widehat{\mathbf{K}} = \alpha \widehat{\mathbf{K}}^{-1} \mathbf{K} \mathbf{R} \mathbf{K}^{-1} \widehat{\mathbf{K}} = \alpha \delta \mathbf{K} \mathbf{R} \delta \mathbf{K}^{-1} = \alpha \widehat{\mathbf{R}} \quad (2.37)$$

où $\delta\mathbf{K} = \widehat{\mathbf{K}}^{-1}\mathbf{K}$ et α est donné par :

$$\alpha = (\det(\widehat{\mathbf{H}}_\infty))^{\frac{1}{3}} \quad (2.38)$$

La matrice d'homographie estimée en présence d'erreurs de calibration peut alors s'écrire :

$$\widehat{\mathbf{H}} = \widehat{\mathbf{H}}_\infty + \widehat{\mathbf{t}}_{df} \widehat{\mathbf{n}}^{fT} \quad (2.39)$$

avec :

$$\widehat{\mathbf{n}}^{fT} = \frac{\mathbf{n}^{fT} \delta\mathbf{K}}{\|\mathbf{n}^{fT} \delta\mathbf{K}^{-1}\|} \quad (2.40)$$

et :

$$\widehat{\mathbf{t}}_{df} = \alpha \|\mathbf{n}^{fT} \delta\mathbf{K}^{-1}\| \delta\mathbf{K} \mathbf{t}_{df} \quad (2.41)$$

Pour plus de précisions sur ce dernier point, le lecteur intéressé pourra se reporter à [Malis 98]. Notons toutefois que la matrice estimée $\widehat{\mathbf{R}} = \frac{1}{\alpha} \widehat{\mathbf{H}}_\infty$ est similaire à la matrice de rotation réelle. Cela implique que les deux matrices ont les mêmes valeurs propres et que les vecteurs propres de $\widehat{\mathbf{R}}$ sont les vecteurs propres de \mathbf{R} multipliés par $\delta\mathbf{K}$. Dans ce cas l'angle de rotation $\widehat{\theta}$ et l'axe de rotation $\widehat{\mathbf{u}}$, obtenus à partir de $\widehat{\mathbf{R}}$, peuvent être exprimés en fonction des valeurs réelles et des erreurs de calibration :

$$\begin{cases} \widehat{\theta} = \theta \\ \widehat{\mathbf{u}} = \frac{\delta\mathbf{K}\mathbf{u}}{\|\delta\mathbf{K}\mathbf{u}\|} \end{cases} \quad (2.42)$$

En pratique, la matrice de colinéation à l'infini n'est pas toujours disponible. On peut cependant l'estimer en utilisant l'algorithme proposé dans [Faugeras 88]. Dans ce cas, les relations (2.40), (2.41) et (2.42) ne sont pas exactes. Comme nous le verrons dans la suite, nos méthodes restent alors très robustes aux erreurs de calibration.

2.4 Conclusion

Dans ce chapitre, nous avons modélisé l'environnement et le capteur caméra en plongeant les espaces de travail dans des espaces projectifs. Nous avons également vu qu'il est possible de remonter à la géométrie épipolaire liant deux images d'une scène statique. Les informations nécessaires à l'estimation de la géométrie épipolaire sont directement extraites de l'image et ne demandent pas de connaissance a priori sur la scène. Cependant, son estimation peut s'avérer délicate, en particulier dans le cas de la matrice fondamentale pour des configurations géométriques particulières (mouvements de rotations pures de la caméra, scène plane \dots). Ces configurations

étant très courantes dans le contexte de l'asservissement visuel, nous utiliserons plutôt la matrice de colinéation, celle-ci ayant l'avantage sur la matrice fondamentale de ne jamais être dégénérée. L'estimation de la matrice de colinéation peut également être problématique dans le cas où les prises de vue sont très rapprochées les unes des autres (en raison de l'influence des bruits de mesure). Ce dernier point peut s'avérer très pénalisant, au voisinage de la position désirée, pour les techniques d'asservissement visuel où un calcul de l'homographie à chaque itération de la boucle de commande est nécessaire. Pour notre part, nous effectuerons l'estimation de la géométrie épipolaire uniquement entre les images initiale et désirée, c'est-à-dire loin de la convergence. Nos méthodes ne seront donc pas pénalisées par cette difficulté. Connaissant la géométrie épipolaire, nous avons également vu qu'il est possible de remonter au déplacement partiel de la caméra et d'obtenir des informations sur la géométrie de la scène. Le cas où la calibration de la caméra est mal connue a également été discuté.

Chapitre 3

L'asservissement visuel

Dans ce chapitre, nous présentons un bref état de l'art des techniques d'asservissement visuel. Nous décrivons les principales structures de commande permettant la réalisation de tâches robotiques à partir des informations visuelles extraites d'images fournies par un capteur de vision. Nous décrivons brièvement l'approche fonction de tâche très largement employée dans ce domaine. Nous présentons ensuite de manière plus précise les différents schémas de commande 2D, 3D et 2D1/2. Nous nous attacherons à préciser les avantages et les inconvénients de ces trois approches.

3.1 Introduction

L'intégration des données capteurs et plus particulièrement de celles issues d'une caméra dans la boucle de commande des robots permet un élargissement important du domaine d'application de la robotique et une amélioration considérable de la précision des positions obtenues. De nombreux travaux ont donc été menés dans ce sens et ont donné lieu à une littérature abondante. Nous invitons le lecteur à consulter [Hutchinson 96, Hashimoto 93a, Corke 93], pour une présentation exhaustive du domaine jusqu'en 1996.

L'objectif de l'asservissement visuel est l'intégration en temps réel d'informations visuelles issues d'une ou plusieurs caméras dans la boucle de commande du robot afin d'en contrôler les mouvements. Les premiers travaux ayant porté sur des systèmes de commande en boucle ouverte sont décrits dans [Rosen 76, Tani 77]. La précision obtenue dépendait alors directement de la justesse des modèles utilisés et de la qualité de la calibration de la caméra et du robot.

Le principe de la boucle fermée a été largement adopté afin d'améliorer la performance des systèmes. Les premiers travaux dans ce sens sont dus à Shirai et Inoue qui décrivent et réalisent une tâche d'assemblage dans [Shirai 73]. Les systèmes de commande par vision peuvent être classifiés, de manière standard, en utilisant deux

critères [Sanderson 80] :

1. l'utilisation ou non d'une boucle interne,
2. l'espace de contrôle.

Dans le cas où le système ne possède pas de boucle interne, le système de vision fournit directement une estimation de l'état du robot aux articulations, se substituant à ses contrôleurs. La stabilisation est uniquement réalisée en utilisant le retour sur le système de vision [Sanderson 83]. Dans ce cas de figure, le système doit fournir une estimation de l'état à une cadence élevée. En raison de l'influence importante des perturbations sur l'estimation temps réel de l'état [Bishop 94], ce type de schéma a été peu utilisé. Weiss exploite cette structure pour contrôler un robot planaire à trois degrés de liberté dont les articulations sont observées par une caméra dans [Weiss 87]. On retrouvera le même type de structure dans [Wijesoma 93] pour un robot à deux axes. Des travaux dans ce sens ont également porté sur la commande de robot à six degrés de liberté [Hashimoto 93b, Gangloff 00].

Dans le cas où le schéma de commande utilisé est hiérarchique (utilisation d'une boucle interne), le système de vision fournit une consigne au contrôleur du robot qui calcule les déplacements à réaliser dans l'espace articulaire par l'intermédiaire d'une boucle fermée interne qui stabilise le système. En utilisant une telle structure, la cadence du processus peut être amenée au temps réel vidéo de 40 ms. Le processus de vision et le système robotique sont alors clairement disjoints, ce qui permet une plus grande portabilité et simplicité des processus de commande par vision. On pourra se reporter à [Corke 96] pour une étude sur les performances dynamiques de cette structure.

Le second point caractérisant les schémas de commande concerne le type d'informations visuelles utilisé pour définir le signal d'erreur à réguler. Dans cette optique, on peut distinguer trois grandes classes de méthodes :

- les asservissements visuels en situation, ou asservissements visuels 3D, où à partir de primitives extraites de l'image, et via un modèle de la cible ou un ensemble d'images de la scène, on détermine la position et l'orientation de la caméra dans un repère de référence. L'erreur entre la configuration désirée et la configuration courante est exprimée dans l'espace de travail du robot;
- les asservissements basés image, ou asservissements visuels 2D où la loi de commande est directement exprimée dans l'espace du capteur;
- les asservissements hybrides ou asservissements visuels 2D1/2 issus des travaux menés à l'IRISA [Malis 98, Malis 99] puis repris ensuite par exemple dans [Morel 00, Camillo 00]. Dans ce cas, l'erreur à réguler se compose à la fois d'informations 3D et 2D.

Nous reviendrons plus en détail sur ces trois approches dans la suite de ce chapitre.

Les systèmes robotisés munis d'une caméra peuvent également être distingués par le positionnement des caméras :

- lorsque la caméra n'a pas de liaison mécanique avec le robot asservi par vision, on parlera d'asservissement visuel déporté. La caméra est positionnée de manière à ce qu'elle observe l'organe terminal [Flandin 00] ou la configuration articulaire du robot [Ruf 00] ou encore à la fois des objets d'intérêt et l'organe terminal du robot dans l'espace de travail [Heraud 98]. Elle peut être fixe ou montée sur un autre système mécanique [Marchand 01].
- lorsque la caméra est montée sur l'organe terminal, on parlera d'asservissement visuel embarqué. Cette configuration est la plus répandue [Hashimoto 93a, Hutchinson 96]. Nous étudierons dans la suite du document ce type de configuration bien que la plupart des résultats puisse être étendue à la configuration déportée.

3.2 La commande en asservissement visuel

3.2.1 Un bref aperçu

De nombreux types de commande peuvent s'appliquer efficacement à des systèmes munis de caméras. Certaines commandes tiennent compte de la dynamique du robot [Rives 97, Koivo 91, Kelly 96, Chuang 97, Tarbouriech 00]. Dans [Gangloff 99, Gangloff 00], une commande de type GPC est appliquée au suivi rapide de profil dans l'espace du capteur dans le cadre d'une architecture de commande directe (sans boucle interne de régulation des positions articulaires du robot). Dans ce travail, l'auteur tient compte à la fois de la dynamique du robot et de la dynamique du capteur.

Une grande partie des travaux se contente d'élaborer des commandes cinématiques et considère le robot comme un intégrateur parfait. Dans [Zanne 00], une commande par mode glissant est appliquée au suivi de trajectoire cartésienne de la caméra. Dans ce cas, la trajectoire de la caméra est simplement définie comme une ligne droite entre sa position initiale et sa position finale. Des commandes par retour d'état non linéaires et par placement de pôles sont proposées dans [Martinet 99] et [Papanikolopoulos 91], respectivement. Des commandes adaptatives sont proposées dans [Weiss 87]. Des commandes de type LQ ou LQG basées sur la minimisation d'un critère quadratique temps-énergie sont décrites dans [Hashimoto 96, Smith 97]. Une commande par retour d'état continu non stationnaire pour des robots non holônomes est proposée dans [Tsakiris 98].

Les approches utilisant une commande de type proportionnel, permettant d'assurer une décroissance exponentielle d'une erreur bien définie, sont les plus courantes

dans le domaine de l'asservissement visuel [Chaumette 90, Hager 97, Hashimoto 97, Martinet 96, Hutchinson 96].

Le formalisme de la fonction de tâche permet de construire de manière simple des lois de commande directement dans l'espace des capteurs. Ce formalisme a été adapté aux problèmes de l'asservissement visuel durant les travaux de thèse décrit dans [Chaumette 90]. Cette technique de commande peut prendre en compte la dynamique du système. Dans la section suivante, nous allons voir comment appliquer cette approche pour établir une commande en vitesse, en négligeant la dynamique du robot, permettant idéalement une décroissance exponentielle d'une erreur bien choisie.

3.2.2 La commande par fonction de tâche

3.2.2.1 Définition de la fonction de tâche

Dans le formalisme développé dans [Samson 91], on considère les cas où les tâches robotiques à réaliser peuvent s'exprimer sous la forme d'une régulation à zéro sur un horizon temporel fini d'une fonction de tâche C^2 , $\mathbf{e}(\mathbf{q}, t)$, qui exprime l'erreur entre la configuration souhaitée et la configuration courante :

$$\mathbf{e}(\mathbf{q}, t) = \mathbf{C}(\mathbf{x}(\mathbf{q}, t) - \mathbf{x}^*(t)) \quad (3.1)$$

où :

- \mathbf{q} représente le vecteur contenant les coordonnées articulaires du robot,
- \mathbf{C} est une matrice de dimension $n \times m$, où n est le nombre de degrés de liberté que l'on désire contrôler et m est la dimension du vecteur de mesure. Cette matrice est appelée **matrice de combinaison** et permet la prise en compte d'une éventuelle redondance d'informations.
- \mathbf{x} est le **vecteur de mesure** obtenu à partir des données visuelles courantes.
- $\mathbf{x}^*(t)$ est la trajectoire désirée de $\mathbf{x}(t)$ (par la suite, on emploiera également le terme **consigne** pour désigner $\mathbf{x}^*(t)$).

Le problème de la régulation de la fonction de tâche est bien posé si \mathbf{e} possède certaines propriétés. En particulier, il doit exister d'une part, une trajectoire idéale et unique de \mathbf{q} telle que la fonction de tâche soit nulle à chaque instant sur l'horizon temporel, et d'autre part, le jacobien de la tâche :

$$\mathbf{J}_e = \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \quad (3.2)$$

doit être régulier autour de cette trajectoire. C'est la propriété d'admissibilité de la fonction de tâche.

Dans le cadre de la commande par vision, il est également nécessaire que les primitives visuelles restent observables sur l'horizon temporel. Dans le cas de l'asservissement 2D, cela implique qu'un nombre suffisant de primitives dans l'image reste dans le champ de vision de la caméra, c'est la condition de visibilité. Dans le cas des asservissements visuels 2D 1/2 et 3D, il faut à la fois que la condition de visibilité soit respectée et que l'estimation des primitives 3D soit réalisable sur l'horizon temporel. Ces conditions ne sont en général pas explicitement prises en compte lors de l'élaboration des lois de commande. Il en résulte que le choix de la fonction de tâche conditionnera significativement l'observabilité.

Le formalisme de la fonction de tâche permet de gérer aisément les degrés de liberté non contraints par la tâche définie par la relation (3.1) afin de réaliser un objectif secondaire. En effet, réaliser la tâche \mathbf{e} (tâche principale) ne nécessite pas toujours l'utilisation de tous les degrés de liberté du robot. Pour utiliser au mieux les fonctionnalités de ce dernier, il est possible de définir une seconde tâche (tâche secondaire) qui est pré-multipliée par l'opérateur de projection orthogonale sur le noyau de \mathbf{J}_e (afin que la seconde tâche n'affecte pas la première). La tâche secondaire peut consister à naviguer autour d'un objet [Berry 00], à éviter les singularités du robot [Marchand 96], les butées articulaires [Marchand 96, Chaumette 00b], les obstacles et les occultations [Marchand 98] ou encore des obstacles [Cadenat 99]. Dans [Flandin 00], une tâche secondaire est utilisée pour contrôler la translation d'une caméra montée sur un bras manipulateur observé par une seconde caméra. Notons cependant que l'utilisation d'une tâche secondaire est possible uniquement lorsque tous les degrés de liberté ne sont pas utilisés pour la réalisation de la tâche principale. Dans ce document, nous introduirons des contraintes directement au niveau de la consigne. Ainsi, nous pourrions traiter les cas où la tâche à réguler nécessite le contrôle de tous les degrés de liberté disponibles.

Comme nous l'avons déjà indiqué, les méthodes d'asservissement visuel peuvent être distinguées par l'espace de contrôle des mouvements de la caméra. Cela se traduit par le choix du vecteur \mathbf{x} dans la fonction de tâche. Après avoir vu comment une commande fournissant le torseur cinématique de la caméra peut être obtenue, nous discuterons ce point plus en détail.

3.2.2.2 Réalisation d'une commande en vitesse

La variation du vecteur d'état \mathbf{x} en fonction du torseur cinématique de la caméra $\boldsymbol{\tau}_c$ joue un rôle central dans l'élaboration de lois de commande en asservissement visuel. Considérons le vecteur d'état \mathbf{x} comme une fonction différentiable de \mathbf{r} (pose de la caméra), sa dérivée peut alors s'écrire de manière générale :

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} + \frac{\partial \mathbf{x}}{\partial t} = \mathbf{L}_x \boldsymbol{\tau}_c + \frac{\partial \mathbf{x}}{\partial t} \quad (3.3)$$

où \mathbf{L}_x exprime le passage entre la vitesse de la caméra et les variations du vecteur d'état \mathbf{x} . Dans le cas où \mathbf{x} est défini à partir de primitives 2D, \mathbf{L}_x est plus connue sous le nom de matrice d'interaction. Pour simplifier l'exposé, on reprendra cette terminologie dans le cas général. De la même manière, on peut définir le jacobien de la tâche \mathbf{J}_t , liant les variations de la fonction de tâche aux variations de la position de la caméra :

$$\frac{\partial \mathbf{e}}{\partial \mathbf{r}} = \mathbf{J}_t \quad (3.4)$$

Si la matrice de combinaison \mathbf{C} ne dépend pas explicitement de \mathbf{r} , la matrice d'interaction et le jacobien de tâche sont liés par la relation suivante :

$$\mathbf{J}_t = \mathbf{C}\mathbf{L}_x \quad (3.5)$$

Considérons maintenant la fonction de tâche \mathbf{e} sous la forme donnée par l'équation (3.1). Si on désire un comportement exponentiel décroissant de constante de temps λ de la fonction de tâche \mathbf{e} , en considérant uniquement la cinématique du robot, cela se traduit par l'équation différentielle d'ordre 1 suivante :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (3.6)$$

Par ailleurs, la dérivée temporelle de la fonction de tâche est donnée par :

$$\dot{\mathbf{e}}(\mathbf{q}, t) = \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{e}}{\partial t} \quad (3.7)$$

que nous pouvons écrire :

$$\dot{\mathbf{e}}(\mathbf{q}, t) = \frac{\partial \mathbf{e}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{e}}{\partial t} \quad (3.8)$$

En outre, le terme $\frac{\partial \mathbf{r}}{\partial \mathbf{q}}$ peut être décomposé en un produit de deux jacobiens :

- le jacobien liant le torseur cinématique de la caméra $\boldsymbol{\tau}_c$ à la vitesse de l'effecteur du robot, celui-ci peut être obtenu lors d'une phase de calibration [Rémy 98, Andreff 01],
- le jacobien exprimant le passage entre la vitesse de l'effecteur du robot et la vitesse articulaire du robot, c'est la matrice jacobienne du robot.

On supposera par la suite que ces deux matrices sont parfaitement connues et inversibles. Notons cependant que nos schémas de commande couplant planification de trajectoires dans l'espace du capteur et asservissement visuel 2D sont robustes vis-à-vis des erreurs de calibration puisque les asservissements visuels dans l'image y sont peu sensibles quand les configurations initiale et désirée sont proches [Espiau 95]. Le jacobien $\frac{\partial \mathbf{r}}{\partial \mathbf{q}}$ sera noté \mathbf{J} et appelé, par abus de langage et soucis de simplicité,

matrice jacobienne du robot. Cette matrice caractérise le passage entre la vitesse de la caméra et la vitesse articulaire du robot :

$$\boldsymbol{\tau}_c = \mathbf{J}\dot{\mathbf{q}} \quad (3.9)$$

Nous considérerons que \mathbf{J} est suffisamment bien connu et on pourra alors se contenter d'une commande définissant le torseur cinématique de la caméra $\boldsymbol{\tau}_c$. La relation (3.8) peut se mettre sous la forme plus compacte suivante :

$$\dot{\mathbf{e}} = \mathbf{J}_t \boldsymbol{\tau}_c + \frac{\partial \mathbf{e}}{\partial t} \quad (3.10)$$

D'après les équations (3.21) et (3.10), le torseur cinématique de la caméra peut s'écrire :

$$\boldsymbol{\tau}_c = -\mathbf{J}_t^{-1} \left(\lambda \mathbf{e} + \frac{\partial \mathbf{e}}{\partial t} \right) \quad (3.11)$$

En pratique, seules des approximations (on les distinguera des valeurs réelles en leur joignant un chapeau) du jacobien de tâche et de $\frac{\partial \mathbf{e}}{\partial t}$ peuvent être utilisées dans la relation précédente, la commande effectivement calculée sera donc :

$$\boldsymbol{\tau}_C = -\widehat{\mathbf{J}}_t^{-1} \left(\lambda \mathbf{e} + \frac{\partial \widehat{\mathbf{e}}}{\partial t} \right) \quad (3.12)$$

En introduisant cette dernière relation dans (3.8), on aboutit à :

$$\dot{\mathbf{e}} = -\mathbf{J}_t \widehat{\mathbf{J}}_t^{-1} \left(\lambda \mathbf{e} + \frac{\partial \widehat{\mathbf{e}}}{\partial t} \right) + \frac{\partial \mathbf{e}}{\partial t} \quad (3.13)$$

Ce qui permet d'obtenir la condition suffisante de décroissance de $\|\mathbf{e}\|$ suivante :

$$\mathbf{J}_t \widehat{\mathbf{J}}_t^{-1} > 0 \Leftrightarrow \mathbf{C} \mathbf{L}_x (\mathbf{C} \widehat{\mathbf{L}}_x)^{-1} > 0 \quad (3.14)$$

On supposera, dans la suite, que l'objet considéré est immobile, ce qui implique :

$$\frac{\partial \mathbf{e}}{\partial t} = \mathbf{C} \frac{\partial \mathbf{x}^*}{\partial t} + \frac{\partial \mathbf{C}}{\partial t} (\mathbf{x} - \mathbf{x}^*) \quad (3.15)$$

où le terme $\frac{\partial \mathbf{x}^*}{\partial t}$ représente la variation temporelle de la trajectoire désirée. Le choix de la trajectoire désirée a une importance primordiale dans le comportement du système et fait l'objet d'une discussion générale dans le paragraphe suivant. Notons également que si la matrice de combinaison choisie ne dépend pas explicitement du temps, la relation (3.15) devient [Berry 99] :

$$\frac{\partial \mathbf{e}}{\partial t} = \mathbf{C} \frac{\partial \mathbf{x}^*}{\partial t} \quad (3.16)$$

3.3 Choix de la consigne

La définition de la consigne $\mathbf{x}^*(t)$ est un problème primordial pour la construction de la fonction de tâche. Les méthodes que l'on retrouve le plus souvent dans la littérature considèrent une consigne fixe, soit $\mathbf{x}^*(t) = \mathbf{x}^*$. Ce choix nécessite l'utilisation de lois de commande stabilisant globalement le système afin d'assurer la convergence de la fonction de tâche. En outre, en présence d'erreur de modélisation et lorsque le déplacement à effectuer est important, les trajectoires obtenues peuvent être surprenantes et bien éloignées de celles attendues. Par conséquent, il est difficile de contrôler les trajectoires réalisées et d'y intégrer des contraintes. Notons également que les propriétés de stabilité globale en présence d'erreur de modélisation n'ont été établies, à notre connaissance, que pour des schémas de commande $3D$ et $2D1/2$ sous certaines hypothèses. Pour le $3D$, dans [Zanne 00] est étudié l'effet des erreurs de calibration sur un algorithme particulier d'estimation de la pose de la caméra [Tsai 86]. Sous l'hypothèse forte que les erreurs de calibration conduisent à une erreur bornée sur la pose, une commande par mode glissant stabilisant le système est obtenue. Dans [Camillo 00], des résultats concernant les bornes de stabilité en présence d'erreurs de calibration de la caméra et du passage effecteur-caméra d'un asservissement visuel $3D$ ont été obtenus sous l'hypothèse que la matrice d'homographie à l'infini soit mesurable durant toute la phase d'asservissement. Dans [Malis 98], la même hypothèse est utilisée pour obtenir les bornes de stabilité des asservissements visuels $2D1/2$.

La consigne (fixe) est généralement obtenue, soit en fixant a priori la pose de la caméra, soit par apprentissage. Dans le premier cas, lorsque des informations $2D$ sont nécessaires, elles sont obtenues en utilisant un modèle précis de la scène et de la caméra en appliquant les équations de projection perspective. Les informations $3D$ sont, quant à elles, fixées par l'utilisateur et obtenues dans divers repères en utilisant des matrices de passage connues. Cette méthode nécessite une connaissance parfaite du système de vision et de l'objet d'intérêt pour obtenir une consigne qui soit physiquement réalisable et pour que celle-ci corresponde bien à l'objectif fixé.

Dans le second cas, la méthode consiste à effectuer un apprentissage expérimental de la consigne. Le robot est d'abord amené dans la configuration désirée et l'image correspondante est acquise. Ensuite le vecteur \mathbf{x}^* est obtenu en appliquant les méthodes utilisées pour le calcul de l'état courant \mathbf{x} . Cette méthode a l'avantage sur la première de fournir une consigne qui correspond effectivement à l'objectif fixé même en présence d'éventuelles erreurs de modélisation. De ce fait, cette méthode est plus adaptée aux problèmes posés en asservissement visuel.

Nous utiliserons la seconde méthode afin d'obtenir l'état désiré du système. Notre objectif sera de fournir une trajectoire adéquate $\mathbf{x}^*(t)$ entre l'état initial et l'état final. On pourra alors se contenter d'une commande localement stable et robuste

puisque l'état courant et l'état désiré restent voisins lors d'un suivi de trajectoire. En outre, nous verrons qu'en utilisant cette approche, il est possible de contrôler efficacement les trajectoires afin d'assurer la convergence du système même lorsque les configurations initiale et désirée sont très éloignées.

Nous allons maintenant distinguer les différents schémas de commande en asservissement visuel selon l'espace dans lequel s'effectue la régulation de la tâche. Nous verrons que les commandes où la fonction de tâche est directement obtenue en utilisant des informations extraites de l'image sont les mieux adaptées à notre problématique.

3.4 Le choix de l'espace de contrôle

3.4.1 L'espace Image

On parlera d'asservissement visuel 2D lorsque le vecteur de mesures \mathbf{x} est uniquement composé d'informations extraites de l'image. Dans ce cas le vecteur \mathbf{x} sera noté \mathbf{s} afin d'utiliser une notation "standard". On considérera également que la configuration des informations visuelles ne dépend que de la situation de la caméra à l'instant t , repérée par un élément $\mathbf{r}(t)$ de $SE(3)$. La fonction de tâche à réguler s'écrit alors :

$$\mathbf{e}(\mathbf{q}, t) = \mathbf{C}(\mathbf{s}(\mathbf{r}(t)) - \mathbf{s}^*(t)) \quad (3.17)$$

Différents types d'informations extraites de l'image peuvent être utilisés. La plupart des travaux en asservissement visuel 2D sont basés sur l'utilisation de coordonnées de points dans les images [Feddemma 89, Chaumette 90, Papanikolopoulos 93, Hager 97]. Il s'avère que l'utilisation de telles primitives peut dans certains cas être problématique [Chaumette 98]. D'autres primitives géométriques peuvent éventuellement être choisies comme des droites dans [Chaumette 90, Andreff 00], des cylindres ou des ellipses dans [Chaumette 90]. Pour appréhender des objets plus complexes, des informations visuelles telles que les moments centraux des surfaces dans [Bien 93, Wells 96], les descripteurs de Fourier dans [Wells 96, Collewet 99], la signature polaire dans [Collewet 99], les invariants projectifs dans [Hager 94] peuvent également être utilisées. Dans [Colombo 99, Drummond 99], les scènes complexes sont traitées en supposant que les pixels de l'image subissent une transformation affine lorsque le robot se déplace (ce qui est une hypothèse assez forte). Dans le cas où des primitives géométriques peuvent difficilement être extraites de l'image, des informations de type mouvement peuvent être choisies [Sundareswaran 94, SantosVictor 97, Crétual 98a, Crétual 98b, Crétual 00].

Dans tous les cas de figure, le comportement du système commandé dépendra fortement des informations visuelles choisies [Hashimoto 98, Chaumette 98], de leur nombre et de leur configuration [Michel 93, Hashimoto 98], mais également du point de vue à partir duquel elles sont observées [Sharma 95, Nelson 96].

Une fois les informations visuelles \mathbf{s} choisies, le calcul de la commande fait intervenir une approximation du jacobien de tâche, donnée ici par :

$$\widehat{\mathbf{J}}_t = \mathbf{C}\widehat{\mathbf{L}}_s$$

où \mathbf{C} est la matrice de combinaison et avec $\mathbf{L}_s = \frac{\partial \mathbf{s}}{\partial \mathbf{r}}$. Cette dernière matrice dépend de \mathbf{s} , mais aussi, en général, des paramètres intrinsèques de la caméra et de certains paramètres géométriques de la scène observée. On trouvera une méthode générale de calcul de cette matrice et son expression pour certaines primitives géométriques dans [Chaumette 90].

Dans la suite du document, nous nous intéresserons à des primitives de type point. Pour un point de coordonnées en pixel $\mathbf{p} = [u \ v \ 1]$, issu de la projection dans le plan image d'un point 3D, \mathcal{M} , de coordonnées $\mathbf{M} = [X_c \ Y_c \ Z_c]^T$ dans le repère de la caméra \mathcal{F}_c , la matrice d'interaction est donnée par :

$$\mathbf{L}_p(Z_c) = \mathbf{a} \begin{bmatrix} -\frac{1}{Z_c} & 0 & \frac{x}{Z_c} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z_c} & \frac{y}{Z_c} & (1+y^2) & -xy & -x \end{bmatrix} \quad (3.18)$$

avec (voir (2.8)) :

$$\mathbf{a} = \begin{bmatrix} \alpha_u & \alpha_{uv} \\ 0 & \alpha_v \end{bmatrix} \quad \text{et} \quad \mathbf{m} = [x \ y \ 1]^T = \mathbf{K}^{-1}\mathbf{p}$$

Quand \mathbf{s} est composé des coordonnées de n points \mathbf{p}^j de l'image, la matrice d'interaction correspondante est donnée par :

$$\mathbf{L}_s(\mathbf{Z}_c) = [\mathbf{L}_{p^1}^T(Z_c^1) \cdots \mathbf{L}_{p^n}^T(Z_c^n)]^T \quad (3.19)$$

où $\mathbf{Z}_c = [Z_c^1 \cdots Z_c^n]^T$.

La loi de commande et sa condition de stabilité font intervenir la matrice de combinaison et une approximation de la matrice d'interaction. En général, le choix de \mathbf{C} est lié au choix de la matrice d'interaction .

Le choix le plus largement adopté consiste à prendre une matrice de combinaison égale à l'identité, soit $\mathbf{C} = \mathbf{I}_m$. Dans ce cas, la condition de stabilité s'écrit :

$$\mathbf{Q} = \mathbf{L}_s \widehat{\mathbf{L}}_s^+ > 0 \quad (3.20)$$

Si on est capable d'estimer suffisamment précisément la valeur courante de la matrice d'interaction, alors on se rapprochera du comportement souhaité donné par la relation (3.21). Dans ce cas, pour que la tâche soit admissible, il faut que le vecteur \mathbf{s} soit de dimension 6. Malheureusement, le choix de six informations 2D,

telles que certaines configurations de la caméra n'entraînent pas une perte de rang de la matrice d'interaction et/ou ne conduisent à un minimum global non désiré de la fonction de tâche, reste un problème ouvert. Dans le cas où le vecteur \mathbf{s} est de dimension supérieure à 6, la fonction de tâche n'est plus admissible. Ce qui implique que certaines trajectoires des primitives dans l'image, imposées par la commande, ne correspondent pas à un déplacement réalisable du robot. L'estimation de la matrice d'interaction, à partir de ce choix pour \mathbf{C} , peut être obtenue de diverses manières :

- en utilisant une formulation par différences finies de la matrice d'interaction et en l'estimant expérimentalement en effectuant une série de mouvements incrémentaux connus dont les répercussions dans l'image sont mesurées [Hosoda 94, Jägersand 97, Suh 93, Domingo 96, Wu 97]. Cette méthode ne nécessite pas de modèle de la cible mais il est impossible d'étudier la stabilité de la commande.
- en calculant la matrice d'interaction, donnée par sa forme analytique, à chaque itération de la boucle de commande. Celle-ci dépendant du type de primitive choisie pour effectuer l'asservissement et fait apparaître des paramètres 3D que l'on peut estimer si le modèle de la cible [Hashimoto 91] ou le déplacement de la caméra [Feddema 90, Papanikolopoulos 94] sont connus. Notons que ce choix peut amener le système à atteindre un minimum local non désiré et/ou une singularité de la matrice d'interaction [Chaumette 98].

Afin de tenir compte de la redondance d'informations, la matrice de combinaison peut être prise égale à une estimation de la pseudo-inverse de la matrice d'interaction à la convergence $\widehat{\mathbf{L}}_{s^*}^+$ [Espiau 92]. Dans ce cas, la tâche est admissible. Comme précédemment, différents choix pour la matrice d'interaction estimée sont alors possibles :

- elle peut être estimée à chaque itération de la boucle de commande de la même manière que pour une matrice de combinaison choisie égale à l'identité;
- elle peut être prise constante et telle que $\widehat{\mathbf{L}}_s = \widehat{\mathbf{L}}_{s^*}$. La trajectoire des primitives dans l'image est alors moins contrainte que dans le cas précédent et ces primitives peuvent éventuellement sortir du champ de vision de la caméra. Le jacobien de tâche estimé $\widehat{\mathbf{J}}_t$ vaut alors l'identité et l'étude de la stabilité revient à l'étude de la positivité du jacobien de tâche réel donné par $\mathbf{J}_t = \widehat{\mathbf{L}}_{s^*}^+ \mathbf{L}_s$. Dans le cas idéal où $\widehat{\mathbf{L}}_{s^*} = \mathbf{L}_{s^*}$, ce type de commande est stable au voisinage de la position désirée et réalise un découplage de la boucle fermée. **Nous nous placerons dans ce cadre** qui semble le mieux adapté au suivi de trajectoire en prenant $\widehat{\mathbf{L}}_{s^*} = \mathbf{L}_{s^*(t)}$. Nous réaliserons alors une commande stable au voisinage de chaque point de la trajectoire planifiée et un découplage de la boucle fermée.

Dans [Hashimoto 93c], la matrice de combinaison est prise égale à la transposée de la matrice d'interaction à la convergence. La matrice d'interaction peut être choisie telle que $\widehat{\mathbf{L}}_s = \mathbf{L}_{s^*}^{T+}$. Le jacobien de tâche estimé $\widehat{\mathbf{J}}_t$ vaut une nouvelle fois l'identité

et l'étude de la stabilité revient à l'étude de la positivité du jacobien de tâche réel donné par $\mathbf{J}_t = \widehat{\mathbf{L}}_{s^*}^T \mathbf{L}_s$. Dans le cas idéal où $\widehat{\mathbf{L}}_{s^*} = \mathbf{L}_{s^*}$, ce type de commande est stable au voisinage de la position désirée car $\mathbf{J}_t = \widehat{\mathbf{L}}_{s^*}^T \mathbf{L}_s > 0$, mais ne réalise pas un découplage de la boucle fermée.

L'étude analytique du domaine de convergence et de la robustesse des lois de commande 2D reste problématique en raison de la complexité des équations mises en jeu dans la condition de stabilité. Cependant, pour certains cas simples, des résultats ont été obtenus dans [Espiau 95]. On peut également observer expérimentalement la convergence de ce type de commande même en présence d'importantes erreurs de modélisation lorsque le déplacement à effectuer n'est pas très important. De ce fait, ce type de commande est très adapté au suivi de trajectoire dans l'espace image.

3.4.2 L'espace Cartésien

Lorsque le vecteur de mesure \mathbf{x} est composé uniquement d'informations caractérisant la situation de la caméra dans l'espace de travail du robot, on parlera d'asservissement visuel 3D. Le contrôle de la caméra est alors effectué dans l'espace cartésien. La relation (3.1) s'écrit :

$$\mathbf{e}(\mathbf{r}, t) = \mathbf{C}(\mathbf{x}(\mathbf{s}(\mathbf{r}), t) - \mathbf{x}^*(t)) \quad (3.21)$$

A l'exception des travaux décrits dans [Martinet 96], où le vecteur \mathbf{x} est composé des coordonnées de points 3D exprimées dans un repère lié à la caméra, les informations \mathbf{x} décrivent la pose de la caméra [Wilson 96, Martinet 96, Martinet 97, Martinet 99, Daucher 97, Rizzi 96, Fagerer 94]. Le vecteur \mathbf{x} est alors une paramétrisation de l'attitude \mathbf{r} de la caméra. Le lien entre les variations de cette paramétrisation et les variations de \mathbf{r} est donné par la matrice d'interaction $\mathbf{L}_x = \frac{\partial \mathbf{x}}{\partial \mathbf{r}}$ avec $\dot{\mathbf{x}} = \mathbf{L}_x \boldsymbol{\tau}_c$.

Six informations suffisent pour caractériser de manière univoque la situation de la caméra dans son environnement. La matrice de combinaison peut donc être choisie égale à l'identité. L'étude de la stabilité de la commande revient alors à l'étude de la positivité de la matrice $\mathbf{L}_x \widehat{\mathbf{L}}_x^+$.

Si on se place dans des conditions idéales, on obtient aisément les conditions de stabilité de la commande en raison de la forme particulièrement simple des matrices d'interaction. Cependant en présence d'erreurs sur les modèles (cible, caméra ...) ou de bruits de mesures, la modélisation de la boucle fermée est très difficile. Cette difficulté majeure apparaît clairement lorsqu'on exprime explicitement la dépendance de la matrice d'interaction vis-à-vis des primitives extraites de l'image \mathbf{s} permettant de remonter au vecteur \mathbf{x} :

$$\mathbf{L}_x = \frac{\partial \mathbf{x}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{r}} = \mathbf{J}_{xs} \mathbf{L}_s \quad (3.22)$$

Le second terme de cette expression n'est rien d'autre que la matrice d'interaction liée à \mathbf{s} et peut en général être obtenu de manière analytique. Par contre, le premier terme est en général inconnu et directement lié à la méthode utilisée pour l'obtention de l'état \mathbf{x} :

- si pour remonter aux informations tridimensionnelles, on utilise une méthode numérique itérative [Yuan 89, Lowe 91, Dementhon 95, Tsai 86] comme dans [Martinet 96], alors il est impossible d'exprimer \mathbf{J}_{xs} et donc d'étudier la stabilité du système en présence d'éventuelles erreurs de modélisation dans le cas général [Chaumette 98]. Pour obtenir certains résultats, il faut alors se placer sous des hypothèses fortes [Zanne 00]. Cela reste vrai pour les méthodes utilisant, pour remonter aux informations tridimensionnelles, le filtrage de Kalman, étendues ou non [Wilson 96, Wunsch 97]. Notons également que ces méthodes nécessitent un modèle 3D de l'objet d'intérêt.
- si pour remonter aux informations tridimensionnelles, on utilise les méthodes issues de la géométrie épipolaire [Faugeras 88] alors, d'une part, le modèle de l'objet n'est plus nécessaire [Basri 98] et, d'autre part, des résultats sur la stabilité du système en présence d'erreurs de modélisation peuvent être obtenus [Camillo 00] mais sous l'hypothèse que la matrice d'homographie à l'infini soit mesurable à chaque itération de la boucle de commande.

En pratique, l'étape de reconstruction des informations tridimensionnelles entraîne une forte sensibilité de ce type d'asservissement aux bruits de mesures. Cela est particulièrement vrai quand la reconstruction est obtenue à partir des matrices fondamentale ou d'homographie, puisque leur estimation est très sensible aux bruits de mesure. Cela peut être très pénalisant près de la convergence. Ce type de schéma semble donc être mal adapté à des suivis de trajectoire, car, dans ce contexte, la mesure courante reste voisine de la situation désirée à chaque instant.

Enfin, le contrôle des mouvements du robot dans son espace de travail permet d'obtenir des trajectoires de l'effecteur très satisfaisantes. En contrepartie, ce type de commande ne permet pas le contrôle des trajectoires des primitives dans l'image et donc celles-ci peuvent très bien sortir du champ de vision de la caméra et ainsi faire échouer l'asservissement.

3.4.3 L'espace Image et l'espace Cartésien

Lorsque le vecteur de mesure \mathbf{x} est composé à la fois d'informations tridimensionnelles et de primitives extraites de l'image, on parlera d'asservissement hybride ou d'asservissement 2D 1/2. Dans ce cas, les informations tridimensionnelles peuvent, comme pour les asservissements 3D, être obtenues en utilisant, soit le modèle de la cible dans une méthode itérative [Cervera 99, Corke 00, Morel 00], soit plusieurs images de la scène (les images courante et désirée) [Malis 99]. Un tel schéma de commande intègre donc à la fois un contrôle de la trajectoire de la caméra dans son

espace de travail et un contrôle de la trajectoire de certaines primitives dans l'image. De cette manière, l'objet d'intérêt a une plus grande probabilité de rester dans le champ de vision de la caméra. En outre, comme dans le cas des asservissements 3D, six informations suffisent pour contrôler les six degrés de liberté de la caméra. La matrice de combinaison peut être choisie égale à l'identité et la condition de stabilité s'écrit $\mathbf{L}_x \widehat{\mathbf{L}}_x^+ > 0$. La matrice d'interaction et donc la condition de stabilité ont une forme simple.

Des résultats formels sur la stabilité et sur la robustesse aux erreurs de modélisation ont pu être obtenus dans le cas où les informations 3D sont estimées à partir de la matrice d'homographie relative à un plan de référence [Malis 98].

Notons cependant que ce type de schéma nécessite l'estimation du déplacement de la caméra à chaque itération de la boucle de commande. Cette phase peut s'avérer délicate pour certaines configurations. Cette étape est, en particulier, pénalisante près de la convergence (en raison de la sensibilité aux bruits de mesure). En outre, même si avec cette méthode, on peut assurer qu'un certain nombre de primitives (en général, une seule primitive visuelle) reste dans l'image, il est impossible de garantir l'observabilité de la tâche durant tout l'asservissement (c'est-à-dire qu'un nombre suffisant de primitives visuelles reste dans l'image pour estimer les paramètres 3D).

3.5 Conclusion

Dans ce chapitre, nous avons présenté les principales techniques permettant la réalisation de tâches robotiques à partir d'informations visuelles.

Les asservissements visuels 3D nécessitent, en général, la connaissance de l'objet observé. L'étape d'estimation des paramètres 3D peut s'avérer délicate en particulier lorsque les paramètres de la caméra ou le modèle de l'objet sont erronés. En outre, la contrainte de visibilité est difficile à intégrer à la structure de commande. En effet, les trajectoires des informations visuelles dans l'image peuvent être inadéquates car les mouvements du robot sont contrôlés directement dans l'espace de travail. Au contraire, le contrôle direct des déplacements du robot dans l'espace Cartésien permet d'obtenir des trajectoires de l'effecteur très satisfaisantes.

Les asservissements visuels hybrides nécessitent également une phase d'estimation de déplacement. Celle-ci est en général réalisée en utilisant plusieurs images de la scène. Dans ce cas, le modèle de la cible n'est pas nécessaire. Comme pour les asservissements 3D, cette étape peut s'avérer délicate lorsque les positions initiale et désirée sont très proches (en raison des inévitables bruits de mesure) et donc près de la convergence. Dans cette technique, une partie du signal d'erreur est directement obtenue dans l'image ce qui permet d'améliorer l'observabilité de la tâche, sans la garantir.

Les techniques d'asservissement visuel 2D permettent une prise en compte plus directe de la contrainte de visibilité car le signal d'erreur à réguler est directement construit à partir d'informations extraites de l'image. Cependant, l'objet d'intérêt peut tout de même quitter l'image durant l'asservissement lorsque les erreurs de modélisation ou le déplacement à effectuer sont importants. En outre, la trajectoire de l'effecteur du robot dans son espace de travail n'est pas du tout contrôlée et celle-ci est alors imprévisible (ou tout du moins sous-optimale) et conduire à un échec de l'asservissement. Lorsque les configurations initiale et désirée sont proches, ce type de méthodes s'avère très efficace. On notera en particulier leur stabilité et robustesse vis-à-vis des erreurs de modélisation dans ce cadre. Dans le contexte du suivi de trajectoires, ce point est primordial, car l'objectif est que la mesure courante reste raisonnablement proche de la consigne courante. L'aspect très local de cette méthode peut donc être pleinement exploité lors d'un suivi de trajectoire.

Nous utiliserons donc les techniques d'asservissement visuel 2D afin de suivre des trajectoires planifiées dans l'image. Nous verrons, de plus, qu'il est possible, lors de la phase de planification de trajectoire, de résoudre les problèmes de sous-optimalité de la trajectoire du robot dans l'espace de travail, et d'introduire certaines contraintes (visibilité, évitement des butées articulaires et des auto-occultations). Nous serons alors en mesure de réaliser, en utilisant un asservissement dans l'image, des tâches de positionnement lorsque le déplacement à effectuer est très important, et donc d'élargir l'aspect très local des techniques actuelles.

Deuxième partie

Planification de trajectoires

Chapitre 4

Introduction et état de l'art

4.1 Introduction

La tâche de base en robotique consiste à amener un robot d'une configuration initiale donnée à une configuration finale désirée. En général, il existe une infinité de trajectoires permettant de réaliser cette tâche. Lorsque la tâche est plus complexe, en raison d'introduction de contraintes, le nombre de chemins possibles reste en général important. L'objectif de la planification est de sélectionner une trajectoire parmi l'ensemble des trajectoires possibles qui assure que certaines contraintes liées à l'environnement, au mécanisme ou aux capteurs utilisés seront satisfaites.

La relation entre la trajectoire (cinématique) du système physique et les forces fournies par les actionneurs générant les mouvements le long de cette trajectoire est donnée par les équations dynamiques du système. En général, la planification consiste donc à la fois à construire la trajectoire cinématique pour le système et à planifier les forces à fournir par les actionneurs le long de la trajectoire. Ces forces peuvent dans certains cas être obtenues directement à partir des trajectoires cinématiques. Dans d'autres cas, la description cinématique du système est utilisée car les équations régissant la dynamique sont très difficiles à obtenir. Le modèle cinématique peut également être utilisé lorsqu'on considère la dynamique du système comme *une boîte noire*. Celle-ci est alors sensée permettre une exécution parfaite de la trajectoire cinématique planifiée. Nous nous placerons dans ce dernier cas de figure. Les mouvements du système physique peuvent être observés et contrôlés dans quatre espaces différents (voir figure 4.1) :

1. **L'espace de tâche** : on appellera espace de tâche, l'espace dans lequel est spécifié la tâche à réaliser et on le notera \mathcal{T} . Cet espace est en général choisi de manière à simplifier les efforts de l'utilisateur lors de la phase de description de la tâche. De ce fait, dans la plupart des applications, c'est un sous-ensemble de l'espace cartésien qui est utilisé. Les coordonnées cartésiennes sont alors utilisées pour spécifier la position dans l'espace et une paramétrisation locale, comme par exemple les angles d'Euler, est choisie pour spécifier l'orientation. De manière générale, on notera \mathbf{x}_c un vecteur qui représente de manière

univoque la position et l'orientation du robot dans l'espace de tâche quand celui-ci est choisi comme étant l'espace cartésien. Nous choisirons dans nos applications l'espace de tâche de manière très différente car nous spécifierons les tâches à réaliser directement dans l'espace du capteur. Plus précisément, nous considérerons que l'espace de tâche est le plan image \mathcal{I} sur lequel la scène observée par la caméra est projetée.

2. **L'espace de travail ou espace opérationnel**: l'espace de travail est un sous-ensemble de l'espace cartésien dans lequel le robot évolue, il sera noté \mathcal{W} dans la suite. L'espace de tâche, quand il est choisi comme un sous-ensemble de l'espace Cartésien, est identique à l'espace de travail, mais peut être paramétrisé de manière différente.
3. **L'espace articulaire**: L'espace articulaire est l'espace dans lequel l'attitude de chacune des articulations du mécanisme est spécifiée. Pour représenter proprement le déplacement d'un système possédant un nombre important de degrés de liberté, il est souvent pratique d'utiliser cet espace.
4. **L'espace des actionneurs**: l'espace des actionneurs est l'espace dans lequel sont spécifiées les commandes à appliquer aux actionneurs pour générer un mouvement du mécanisme. Un actionneur est en général contrôlé par une entrée unique (par exemple un courant ou une tension dans le cas des actionneurs électromécaniques). Typiquement, l'espace des actionneurs sera donc un sous ensemble de \mathbb{R}^m , où m représente le nombre d'actionneurs.

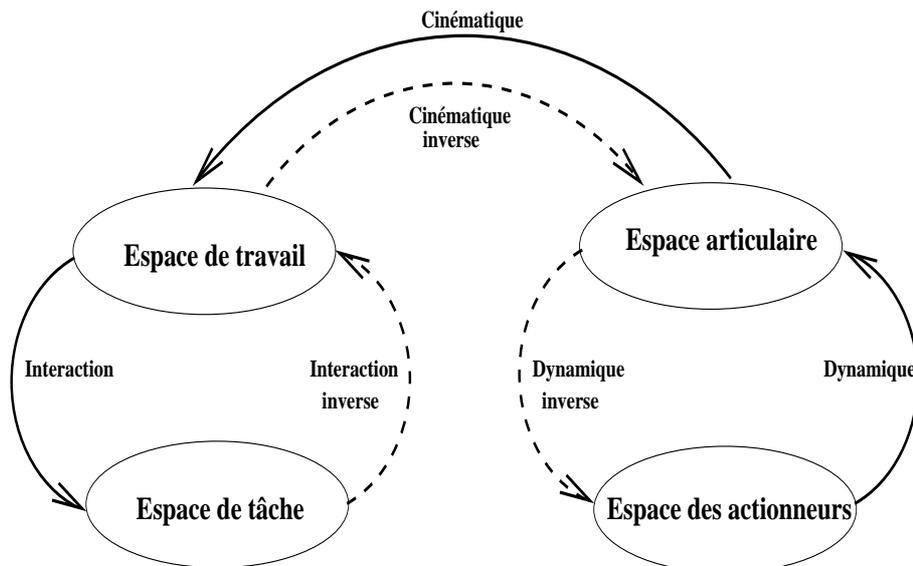


FIG. 4.1 – *Espaces dans lesquels peuvent être observés les mouvements du robot et les applications qui les relient*

Les applications entre l'espace de tâche, l'espace de travail, l'espace articulaire et l'espace des actionneurs ne sont pas toujours inversibles. Par exemple, dans le cas d'un robot dont la cinématique est redondante, il y a une application entre l'espace articulaire et l'espace de travail, mais une trajectoire dans l'espace de travail ne spécifie pas de manière univoque une trajectoire dans l'espace articulaire. En général, une courbe dans l'espace des actionneurs associée à un ensemble de conditions aux limites est représentée de manière unique dans l'espace articulaire, dans l'espace de travail et dans l'espace de tâche. L'application entre l'espace des actionneurs et l'espace articulaire est donnée par les équations du mouvement du mécanisme et entraîne la manipulation d'un ensemble d'équations différentielles. Au contraire, l'application entre l'espace articulaire et l'espace de travail est algébrique. L'application entre l'espace de travail et l'espace de tâche est également de nature algébrique :

- dans le cas où l'espace de tâche correspond à l'espace de travail, (i.e. à un sous-ensemble de l'espace cartésien) alors il ne s'agit que d'un changement de paramétrisation;
- dans le cas où l'espace de tâche correspond à l'espace image, alors les relations mises en jeu relèvent de la géométrie projective et plus particulièrement de la géométrie épipolaire.

Cela implique qu'une trajectoire dans l'espace des actionneurs caractérise complètement les mouvements du robot dans tous les autres espaces. Cependant pour la majeure partie des applications, l'obtention de la trajectoire dans l'espace des actionneurs est difficile et nécessite une modélisation parfaite du mécanisme, du capteur et de l'environnement. On préfère plutôt générer une trajectoire dans l'espace articulaire et/ou dans l'espace de travail et/ou dans l'espace de tâche. Cette trajectoire peut alors être suivie précisément en utilisant une loi de commande adaptée. Dans nos applications, nous chercherons à planifier les mouvements du robot dans l'espace de tâche choisi comme l'espace image. La trajectoire dans l'espace de tâche est déduite d'une trajectoire dans l'espace de travail, préalablement planifiée. Planifier directement dans l'image nécessiterait d'introduire la contrainte que les images obtenues correspondent à des positions atteignables de la caméra. Résoudre ce problème inverse est une tâche extrêmement complexe. C'est pourquoi nous avons choisi de planifier la trajectoire de la caméra en 3D puis de calculer par simple projection les positions associées dans le plan image. La trajectoire dans l'espace de travail permettra également d'obtenir une trajectoire dans l'espace articulaire. Puisque la transformation entre l'espace de travail et l'espace articulaire fournit, en général, un ensemble de solutions, nous choisirons la solution obtenue en utilisant l'inverse généralisée de Moore-Penrose de cette transformation. Enfin, nous ne considérerons pas les éventuels problèmes liés au passage de l'espace articulaire à celui des actionneurs.

Dans le cadre de l'asservissement visuel, une telle approche comporte de nombreux avantages. En premier lieu, la spécification complète de trajectoires de la

caméra dans l'espace de travail assure que les trajectoires des primitives visuelles dans l'image sont physiquement réalisables, ce qui permet la construction d'une fonction de tâche admissible. Si les trajectoires dans l'image sont obtenues à partir d'une trajectoire discrète de la caméra, une phase d'interpolation dans l'image sera mise en œuvre afin d'obtenir des trajectoires continues et dérivables. Cela permet d'améliorer le comportement de la commande et de gérer aisément le calcul des consignes. En outre, comme nous le verrons dans la suite, il est possible d'introduire lors de la phase de planification des contraintes liées au type de capteur utilisé, à l'environnement ou aux mécanismes et de prendre en compte des critères d'optimalité. Comme nous l'avons précisé précédemment, les trajectoires sont au préalable planifiées dans l'espace de travail. Une commande de type 2D1/2 ou 3D pourrait alors être utilisée pour suivre ces trajectoires. Nous préférons cependant planifier les trajectoires de primitives visuelles dans l'image et utiliser un asservissement visuel 2D. En effet, nous démontrerons dans la suite que les trajectoires dans l'image ne sont pas influencées par des erreurs sur la calibration de la caméra (contrairement aux trajectoires de la caméra dans l'espace de travail). En outre, cela nous permettra d'exploiter pleinement les avantages des asservissements visuels 2D. D'une part, l'estimation à chaque itération de la boucle de commande de la situation de la caméra ne sera pas nécessaire. D'autre part, en raison du bouclage de la commande directement dans l'image, les asservissements 2D sont très robustes aux différentes erreurs de modélisation et aux bruits de mesure lorsque les configurations initiale et désirée sont voisines et sont donc bien adaptés au suivi de trajectoires.

Les travaux dans le domaine de la planification de trajectoires pour les robots holônômes ou non sont considérables. Nous ne parcourons que très brièvement la littérature de ce domaine par la suite.

4.2 Génération de trajectoires en robotique

Un concept important qui a bénéficié des travaux en planification cinématique de mouvement est la notion d'espace de configuration. Il a été défini dans le domaine de la robotique par Lozano-Pérez et Wesley dans [LozanoPerez 79]. L'idée de base est de représenter le robot par un point dans un espace approprié, appelé espace de configuration et que l'on notera \mathcal{C} . Chaque point de cet espace correspond à une configuration différente du robot. Dans cet espace, une contrainte géométrique correspond à un ensemble de configurations interdites pour le robot, cet espace sera noté \mathcal{C}_{obs} . L'ensemble des configurations admissibles (appelé également *espace libre*) sera noté \mathcal{C}_{free} . La séparation de ces deux sous-espaces est en général réalisée en utilisant l'une des deux méthodes suivantes :

- l'espace de configuration du robot \mathcal{C} est divisé en cellules, chaque cellule appartient soit au sous-espace accessible au robot \mathcal{C}_{free} , soit au sous-espace interdit au robot \mathcal{C}_{obs} .
- Le contour des obstacles sert à séparer l'espace libre et l'espace interdit.

Le problème de base de la planification de trajectoires cinématique peut alors être formulé de la façon suivante [Latombe 91] :

Le problème de base en planification cinématique. *Étant donné une position et une orientation initiales ainsi qu'une position et une orientation désirées du robot dans l'espace libre du robot C_{free} , générer un chemin σ spécifiant une séquence continue de positions et d'orientations du robot évitant C_{obs} , partant de la position et de l'orientation initiales et terminant dans la position et l'orientation finales.*

Une première classification des méthodes permettant de résoudre le problème de base de la planification de trajectoires en robotique peut être faite selon que la stratégie adoptée utilise une représentation locale du système ou une représentation globale. Il existe néanmoins de nombreux travaux sur le couplage de ces deux approches [Barraquand 92].

4.2.1 Les méthodes locales

Les méthodes locales ne nécessitent qu'une perception de l'environnement immédiat du robot. La plupart des stratégies locales se fondent sur une approche où un capteur, comme un sonar ou une caméra, fournit des informations sur le voisinage immédiat du robot, qui sont intégrées à un ensemble de règles hiérarchiques fournissant le comportement à adopter. De tels schémas prennent rarement en compte les critères d'optimalité dans la recherche du chemin à suivre. Ces méthodes locales utilisent la connaissance courante de l'état du système et de l'environnement pour en déduire le mouvement suivant. Ces stratégies ne nécessitant pas de calcul a priori, elles sont particulièrement appropriées aux tâches dont le seul objectif est d'atteindre une configuration désirée sans contrainte sur la trajectoire à réaliser. **La méthode des fonctions de potentiel** est probablement la plus utilisée pour la planification de trajectoires en ligne. Cette méthode est basée sur la construction de champs de potentiel artificiels qui peuvent être attractifs ou répulsifs selon qu'ils sont associés à l'objectif ou à un obstacle de l'environnement. A chaque instant, le robot se déplace dans la direction opposée au gradient d'une somme pondérée des potentiels qui est analogue à une force artificielle appliquée au robot. L'idée des champs de potentiel est en premier lieu apparue dans le document de thèse de Khatib [Khatib 80] et fut formellement utilisée dans le contexte de la planification de trajectoires dans [Khatib 86]. Le défaut majeur d'une telle approche réside dans le fait que des minima locaux peuvent éventuellement faire échouer le processus de planification. Koditschek, dans [Koditschek 87], utilise le théorème de Poincaré-Hopf pour montrer qu'une fonction de potentiel ne possédant qu'un seul point d'équilibre n'existe pas dans le cas général. Plus précisément, dans \mathbb{R}^2 , s'il y a q obstacles homéomorphiques à un disque fermé de rayon 1, alors la fonction de potentiel doit avoir au moins q points selles. Rimon et Koditschek proposent pour contourner cette difficulté de construire des fonctions de potentiels appelées fonctions de navigation,

possédant un minimum global, et pour lesquelles tous les autres points d'équilibre sont instables [Rimon 92]. Cependant, la construction des fonctions de navigation nécessite une connaissance parfaite de la topologie de l'espace de configuration et donc une grande partie des avantages de la méthode des potentiels est perdue. Un autre inconvénient de la méthode des potentiels est que, de manière générale, les trajectoires obtenues sont loin d'être optimales. Nous décrirons plus en détail cette méthode dans le cadre de nos travaux sur la planification de trajectoires dans l'image.

4.2.2 Les méthodes globales

Dans un premier temps, nous décrirons les algorithmes développés pour résoudre le problème de planification de trajectoires dans un espace discrétisé. La discrétisation permet de réduire le problème de la recherche d'une solution dans l'espace de dimensions infinies des solutions possibles à un espace de recherche de dimensions finies. Un des avantages des algorithmes discrets est qu'ils peuvent répondre à la question de complétude : si des solutions existent dans l'espace réduit, l'algorithme en retournera une, s'il n'y a pas de solution, l'algorithme le signifiera. Ces stratégies tiennent compte des propriétés globales de l'espace de configuration du robot et peuvent intégrer des contraintes spécifiques, qui doivent être satisfaites à chaque étape nécessaire à la réalisation de la tâche. Ces méthodes sont très intéressantes du point de vue de la commande. En effet, une fois que la trajectoire du système est planifiée, le système peut être linéarisé le long de cette trajectoire et les méthodes issues de la théorie de la commande des systèmes linéaires peuvent être utilisées pour contrôler les mouvements du robot autour de la trajectoire nominale [Walsh 94]. Cependant ces approches ont l'inconvénient de s'adapter difficilement aux éventuelles incertitudes concernant le déroulement de la tâche. En outre, elles sont beaucoup plus coûteuses en terme de temps de calcul que les méthodes locales. Une excellente description des méthodes de planification de la cinématique du déplacement des robots est faite dans le livre de Latombe [Latombe 91] qui fait référence dans le domaine. Latombe y distingue trois types de méthodes : les approches par *roadmap*, les approches par décomposition en cellules et les méthodes par fonctions de potentiel que nous avons brièvement décrites dans la partie précédente.

La méthode des roadmap : dans cette approche un réseau de courbes, appelé *roadmap*, est défini comme un ensemble de courbes monodimensionnelles capturant la connectivité de l'espace libre du robot \mathcal{C}_{free} [Canny 88]. Une fois que la *roadmap* \mathcal{R} est construite, elle est utilisée comme un ensemble de chemins "standards". La tâche suivante consiste alors à trouver un chemin approprié entre les points initial et final dans \mathcal{R} . La difficulté principale dans cette méthode se situe au niveau de la construction de \mathcal{R} . De nombreuses méthodes ont été proposées pour obtenir \mathcal{R} . On citera par exemple les très populaires graphes de visibilité [Laumond 87], la méthode des cônes généralisés ("freeway net") [Brooks 83] ou encore la méthode des

silhouettes [Canny 88].

La décomposition en cellules est probablement la méthode la plus populaire dans le domaine de la planification de trajectoires et celle qui a donné lieu à la plus importante littérature [Avnaim 88, Zhu 91]. Basiquement, l'objectif est dans un premier temps de diviser l'espace libre du robot \mathcal{C}_{free} en parcelles appelées cellules, de telle sorte que le chemin entre deux configurations dans une cellule soit aisément généré. Un graphe représentant les relations d'adjacence entre les cellules est ensuite construit. Ce graphe est communément appelé graphe des connectivités. Le problème se réduit alors à la recherche dans le graphe des connectivités d'un chemin "optimal" selon un critère bien défini.

Dans le chapitre 5 nous utiliserons également des méthodes continues de planification de trajectoires. Dans la partie suivante, nous allons brièvement décrire ces méthodes.

4.2.3 Les méthodes continues

Contrairement aux méthodes discrètes, pour les méthodes continues la question de complétude est complexe et en général ne peut pas être résolue. Le problème est mis sous la forme de la minimisation d'une fonction de coût le long de la trajectoire d'un système dynamique décrit par un ensemble d'équations différentielles. Une grande partie de ces méthodes est basée sur la théorie de la commande optimale [Athans 66] et du calcul variationnel [Sagan 69]. Buckley considère la minimisation d'une fonction de coût sous forme intégrale en utilisant le calcul variationnel [Buckley 85]. Il résout de cette manière un grand nombre de problèmes non contraints de planification de trajectoire dans un plan. Ce type d'approche a également été utilisé pour l'obtention de trajectoires au sens du plus court chemin pour les robots non holônomes [Souères 97, Laumond 94]. La théorie de la commande optimale a été exploitée pour la commande et la planification de trajectoires à temps minimal. L'objectif est alors de minimiser la durée du déplacement du robot entre les configurations initiale et désirée en tenant compte des capacités du mécanisme. Les premiers travaux dans ce sens sont dus à Kahn et Roth [Kahn 71]. Ce type d'approche a ensuite été très largement repris, par exemple pour obtenir une trajectoire dans l'espace des actionneurs quand la trajectoire dans l'espace articulaire est obtenue indépendamment [Bobrow 85, Slotine 89], ou pour obtenir une trajectoire à temps minimal dans un espace encombré [Shiller 91]. Bien que l'étude théorique des solutions de la planification de trajectoires à temps minimal soit attractive, d'un point de vue pratique, les applications restent limitées. Il s'avère, en effet, que ces solutions conduisent à des discontinuités dans l'espace des actionneurs. Afin de pallier ce problème, des travaux se sont focalisés sur la définition et l'utilisation de fonction de coût prenant en compte une mesure de continuité des mouvements. Une possibilité est de minimiser une fonction de coût qui prenne en

compte à la fois la durée du déplacement et l'intégrale de la norme au carré du vecteur de commande. Singh et Leu utilisent la programmation dynamique pour résoudre ce problème [Singh 89], alors que Bessonnet et Lallemand utilisent le principe du minimum de Pontryagin [Bessonnet 94]. Pour des applications où la durée du déplacement et les limitations des actionneurs ne sont pas fondamentales, des fonctions de coût de nature différente peuvent être utilisées. On pourra par exemple se contenter de la norme au carré du vecteur de commande [Vutobratović 82] ou de la norme au carré du torseur cinématique [Nakamura 87]. Dans nos applications, nous utiliserons comme mesure de continuité, l'intégrale de la norme au carré du vecteur de commande ou de sa dérivé temporelle; le vecteur de commande étant, dans notre cas, le torseur cinématique de la caméra.

Les premiers résultats concernant la commande optimale en présence de contraintes inégalités sont dus à Bryson et *al* [Bryson 63]. La méthode est basée sur une modification des solutions du problème de Bolza, sur lequel nous reviendrons plus longuement dans le chapitre 5. La reformulation des problèmes de contrôle optimal sous forme variationnelle a été appliquée avec succès pour la planification de trajectoires de robot holonome [Buckley 85] ou non [Desai 99]. L'approche que nous adopterons est similaire à celle utilisée dans [Desai 99, Zefran 97]. Bien que ces problèmes puissent être traités en utilisant une minimisation locale et dans un espace de dimension finie [Martin 89, Suh 87], nous le décrirons sous l'angle d'une approche globale et dans un espace des solutions infinies [Zefran 97].

Nous terminons ce bref état de l'art sur la planification de trajectoires en robotique par les méthodes probabilistes.

4.2.4 Les méthodes probabilistes

Les algorithmes déterministes complets pour la planification de trajectoires (i.e. pouvant répondre à la question de complétude) sont très coûteux en temps de calcul quand la dimension de l'espace de configuration devient important. Des méthodes permettant de répondre à des problèmes complexes peuvent être appliquées pour de grandes dimensions de l'espace de configuration, en relaxant la notion de complétude au bénéfice d'une efficacité pratique et d'un complétude probabiliste. Un planificateur sera dit probabilistiquement complet, si étant donné un problème résolvable, la probabilité d'obtenir la solution tend vers 1 lorsque le temps de calcul tend vers l'infini. Les deux approches les plus efficaces sont les *probabilistic path planner* et *randomized path planner*. Nous ne développerons pas plus ces approches et renvoyons le lecteur intéressé à [Barraquand 91, Langlois 92] pour la première et à [Overmars 94, Kavraki 96] pour la seconde.

4.3 La planification de trajectoires dans l'image

Bien que la littérature du domaine de l'asservissement visuel soit très importante, peu de travaux abordent la planification de trajectoires dans l'espace image. Cette phase de planification est pourtant très intéressante dans ce contexte. Dans ce cadre, le contrôle du système s'effectue en général dans l'espace de tâche qui est choisi, soit comme étant l'espace de travail, soit comme étant l'espace image. Il est alors très difficile d'introduire des contraintes spécifiques à chaque espace (visibilité dans l'image, optimalité des trajectoires dans l'espace cartésien, butées mécaniques dans l'espace articulaire). En outre, comme nous l'avons déjà mentionné, les schémas de commande 2D sont très stables et robustes vis-à-vis d'erreurs de modélisation et des bruits de mesure lorsque le déplacement à effectuer est suffisamment petit. Ils sont donc bien adaptés à un suivi de trajectoires.

Dans un cas très simple où l'espace de travail est supposé planaire et l'espace image linéaire, Cowan et Koditschek proposent, dans [Cowan 99], une méthode globalement stable en utilisant le principe des fonctions de navigation [Rimon 92]. Dans [Hosoda 95], un planificateur de trajectoires dans l'image utilisant une cible connue et un système stéréo est proposé et appliqué à l'évitement d'obstacles. Une tâche de positionnement utilisant des vues intermédiaires synthétisées par morphing est présentée dans [Singh 98]. Ruf et *al* proposent un planificateur de trajectoires pour un robot en translation dont les mouvements sont observés par une paire de caméra faiblement calibrée [Ruf 97]. Dans [Borgstadt 01], la trajectoire dans l'image d'un ensemble de primitives 3D connues est obtenue par décomposition sous une forme canonique de la transformation projective. Dans tous ces travaux, les contraintes spécifiques à chaque espace et les problèmes de robustesse ne sont pas abordés.

Nos travaux ont pour cadre la planification de trajectoires dans l'espace de tâche quand celui-ci est choisi comme l'espace image. Nous reformulerons donc le problème de base de la planification de trajectoires afin de tenir compte de cette spécificité :

Le problème de base en planification de trajectoires dans l'image : *étant donné une image initiale et une image finale d'un objet d'intérêt, fournies par une caméra, générer un chemin σ spécifiant une séquence continue d'images de l'objet d'intérêt correspondant à des attitudes physiquement atteignables par le robot, partant de l'image initiale et terminant à l'image désirée.*

Ce problème de base peut être complété en intégrant des notions d'optimalité et/ou des contraintes spécifiques aux différents espaces concernés. On s'intéressera à la qualité de la trajectoire de la caméra dans l'espace de travail afin que celle-ci soit optimale vis-à-vis de critères que nous préciserons par la suite. Nous verrons également comment intégrer des contraintes à la phase de planification. En particulier nous

nous intéresserons à l'intégration de contraintes dans l'espace image (visibilité de l'objet d'intérêt) et dans l'espace articulaire (butées mécaniques).

Dans le chapitre suivant, nous utilisons le formalisme de la commande optimale afin d'obtenir des trajectoires de primitives visuelles extraites de l'image qui correspondent à un déplacement géométrique et/ou temporel optimal de la caméra dans l'espace de travail du robot.

Chapitre 5

Génération de trajectoires à minimum d'énergie ou d'accélération

Dans ce chapitre, nous montrons comment, à partir de deux images d'une scène statique inconnue acquises par une caméra, une trajectoire continue dans l'image de primitives visuelles correspondant à une trajectoire optimale (géométrique et temporelle) de la caméra peut être obtenue pour relier ces deux images. Nous résolvons les problèmes à minimum d'énergie et minimum d'accélération de manière analytique. Comme attendu, nous verrons que les trajectoires géométriques de la caméra sont des géodésiques et que seule la paramétrisation temporelle est différente selon le critère d'optimalité choisi. Ces trajectoires correspondent à des trajectoires particulières des matrices de colinéation que nous utilisons afin d'obtenir les trajectoires des primitives visuelles dans l'image. Nous montrons que ces dernières sont indépendantes de la calibration de la caméra. Nous utilisons ensuite un schéma de contrôle de type 2D pour suivre les trajectoires dans l'image. Nous étudions également le cas de l'interpolation d'un nombre quelconque d'images afin de pallier le problème de mise en correspondance des primitives visuelles, lorsque les prises de vue initiale et finale sont distantes. Ensuite, à titre indicatif, nous mettons en œuvre une méthode numérique réalisant une optimisation globale afin d'introduire la contrainte de visibilité et en décrivons les limites.

5.1 Etude du cas sans contrainte de visibilité

Dans la suite, nous considérons qu'étant donné une image initiale et une image désirée de la scène, des primitives visuelles (points, segments, contours) peuvent être extraites et mises en correspondance dans les deux images. A partir de ces primitives, la matrice de colinéation, \mathbf{G}_0 , relative à un plan Π de référence à l'instant $t = 0$ et la matrice de colinéation à l'infini $\mathbf{G}_{0\infty}$ (afin de démontrer l'indépendance vis-à-vis de

la calibration de la caméra) sont estimées. Rappelons que lorsque la configuration désirée est atteinte, à l'instant $t = 1$, la colinéation est proportionnelle à la matrice identité de dimension 3×3 : $\mathbf{G}^f \propto \mathbf{I}$.

Dans les paragraphes suivants, nous cherchons une trajectoire continue $\mathbf{G}(t)$ de la matrice de colinéation relative au plan Π , correspondant à un chemin optimal de la caméra et telle que $\mathbf{G}(0) \propto \mathbf{G}_0$ et $\mathbf{G}(1) \propto \mathbf{G}^f$. Deux critères d'optimalité seront utilisés :

- l'intégrale de la norme au carré de la dérivé temporelle de la position de la caméra sur l'horizon temporel (problème à minimum d'énergie),
- l'intégrale de la norme au carré de la dérivée seconde de la position de la caméra sur l'horizon temporel (problème à minimum d'accélération).

Nous traitons dans un premier temps le cas où le mouvement de la caméra est une rotation pure, puis le cas où ce déplacement est une translation pure. Finalement nous considérons le cas où la caméra effectue un déplacement rigide dans l'espace cartésien. Nous utilisons ensuite les résultats obtenus afin d'exprimer la trajectoire de primitives visuelles (en considérant que les modèles des primitives 3D correspondants sont inconnus) dans l'image sous forme analytique.

5.1.1 Mouvement de rotation pure

Lorsque le mouvement de la caméra est une rotation pure la matrice de colinéation est similaire à un facteur d'échelle près à une matrice de rotation :

$$\mathbf{G} \propto \mathbf{K}\mathbf{R}\mathbf{K}^{-1} \quad (5.1)$$

Nous chercherons donc une trajectoire continue de \mathbf{G} , lorsque celle-ci peut se mettre sous la forme précédente. Nous rappelons dans un premier temps, quelques propriétés du groupe des rotations qui nous seront utiles dans la suite.

5.1.1.1 Quelques éléments sur le groupe des rotations

Le groupe des rotations $SO(3)$ est l'ensemble des matrices réelles orthogonales dont le déterminant est égal à 1. Le groupe $SO(3)$ a la structure d'un groupe de Lie. Dans un groupe de Lie, l'espace tangent à l'identité a la structure d'une algèbre de Lie. L'algèbre de Lie de $SO(3)$ est notée $so(3)$. Elle est composée des matrices antisymétriques de dimension 3×3 , de la forme :

$$[\boldsymbol{\theta}]_{\times} = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}$$

Une des connexions les plus importantes entre un groupe de Lie et son algèbre est la fonction exponentielle. Pour tout élément \mathbf{R} de $SO(3)$, il existe au moins un

élément $[\boldsymbol{\theta}]_{\times}$ de $so(3)$ tel que $e^{[\boldsymbol{\theta}]_{\times}} = \mathbf{R}$. Cela peut s'écrire en utilisant la formule de Rodrigues :

$$\mathbf{R} = e^{[\boldsymbol{\theta}]_{\times}} = \mathbf{I} + \frac{\sin \|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|} [\boldsymbol{\theta}]_{\times} + \frac{1 - \cos \|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|^2} [\boldsymbol{\theta}]_{\times}^2 \quad (5.2)$$

où $\|\boldsymbol{\theta}\|$ est la norme euclidienne standard. Inversement, si \mathbf{R} est un élément de $SO(3)$ tel que $\text{Trace}(\mathbf{R}) \neq -1$ alors :

$$\log(\mathbf{R}) = \frac{\theta}{2 \sin \theta} (\mathbf{R} - \mathbf{R}^T) \quad (5.3)$$

où θ vérifie :

$$\theta = \|\boldsymbol{\theta}\| = \arccos \left(\frac{1}{2} (\text{Trace}(\mathbf{R}) - 1) \right) \quad (5.4)$$

Si $\text{Trace}(\mathbf{R}) = -1$, $\log(\mathbf{R})$ peut être obtenu en notant que $\boldsymbol{\theta} = \mathbf{u}\theta$ où \mathbf{u} est le vecteur propre de norme 1 de \mathbf{R} associé à la valeur propre 1.

Une autre connexion importante entre $so(3)$ et $SO(3)$ met en jeu les vitesses angulaires. Si $\mathbf{R}(t)$ est une courbe de $SO(3)$, alors $\dot{\mathbf{R}}\mathbf{R}^T$ et $\mathbf{R}^T\dot{\mathbf{R}}$ sont des matrices anti-symétriques et donc des éléments de $so(3)$. $\mathbf{R}^T\dot{\mathbf{R}}$ est la matrice antisymétrique associée au vecteur vitesse angulaire $\boldsymbol{\omega}$:

$$\mathbf{R}^T\dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times} \quad (5.5)$$

5.1.1.2 Trajectoire optimale de la colinéation

Nous considérons ici les cas où la caméra effectue uniquement des mouvements de rotation pour atteindre la position désirée. Nous utilisons deux fonctions de coût différentes afin d'obtenir les solutions des problèmes à énergie minimale (problème *PR1*) et à accélération minimale (problème *PR2*).

Problème PR1. On cherche dans ce cas la trajectoire de la matrice de colinéation qui correspond au minimum d'énergie. Ce problème peut s'écrire sous la forme suivante :

Trouver $\mathbf{G}(t)$ minimisant $J_1 = \int_0^1 \boldsymbol{\omega}^T \boldsymbol{\omega} dt$

avec :

$$\mathbf{R}^T\dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times},$$

$$\mathbf{G} \propto \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$$

et les conditions aux limites suivantes :

$$\mathbf{G}(0) \propto \mathbf{G}_0, \quad \text{et} \quad \mathbf{G}(1) \propto \mathbf{I}$$

Notons que les conditions aux limites du problème *PR1* sont satisfaites pour les valeurs initiale et finale suivantes de la matrice de rotation :

$$\mathbf{R}(0) = \mathbf{R}_0 \quad \text{et} \quad \mathbf{R}(1) = \mathbf{I}$$

Lorsqu'on substituera ces nouvelles conditions aux limites à celles du problème initial, on renommera *PRN1* le problème *PR1*. En utilisant ces conditions aux limites le problème *PR1* peut se décomposer en deux étapes :

1. trouver $\mathbf{R}(t)$ minimisant $J_1 = \int_0^1 \boldsymbol{\omega}^T \boldsymbol{\omega} dt$ avec $\mathbf{R}^T \dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times}$ et les conditions aux limites $\mathbf{R}(0) = \mathbf{R}_0$ et $\mathbf{R}(1) = \mathbf{I}$;
2. en déduire $\mathbf{G}(t)$ sachant $\mathbf{G}(t) \propto \mathbf{K}\mathbf{R}(t)\mathbf{K}^{-1}$.

La difficulté consiste à trouver la solution au premier point. Celle-ci peut être obtenue en utilisant le principe du minimum de Pontryagin (voir annexe B).

Problème PR2. Nous cherchons ici la trajectoire de la matrice de colinéation qui correspond au problème du minimum d'accélération. Comme pour le problème *PR1*, nous formulons ce problème de manière standard :

Trouver $\mathbf{G}(t)$ minimisant $J_2 = \int_0^1 \dot{\boldsymbol{\omega}}^T \dot{\boldsymbol{\omega}} dt$
avec :

$$[\boldsymbol{\omega}]_{\times} = \mathbf{R}^T \dot{\mathbf{R}}$$

$$\mathbf{G} \propto \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$$

et les conditions aux limites suivantes :

$$\mathbf{G}(0) \propto \mathbf{G}_0$$

$$\mathbf{G}(1) \propto \mathbf{I}$$

$$\boldsymbol{\omega}(0) = \mathbf{0}$$

$$\boldsymbol{\omega}(1) = \mathbf{0}$$

De la même manière que pour le problème *PR1*, notons que les conditions aux limites du problème *PR2* sont vérifiées pour :

$$\mathbf{R}(0) = \mathbf{R}_0; \quad \mathbf{R}(1) = \mathbf{I}; \quad \boldsymbol{\omega}(0) = \mathbf{0}; \quad \boldsymbol{\omega}(1) = \mathbf{0}.$$

Ce problème sera noté dans la suite *PRN2*. Le problème *PRN2* peut également être décomposé en deux étapes :

1. trouver $\mathbf{R}(t)$ minimisant $J_2 = \int_0^1 \dot{\boldsymbol{\omega}} \dot{\boldsymbol{\omega}}^T dt$ avec $\mathbf{R}^T \dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times}$ et les conditions aux limites $\mathbf{R}(0) = \mathbf{R}_0$, $\mathbf{R}(1) = \mathbf{I}$, $\boldsymbol{\omega}(0) = \mathbf{0}$ et $\boldsymbol{\omega}(1) = \mathbf{0}$,

2. en déduire $\mathbf{G}(t)$ sachant $\mathbf{G}(t) \propto \mathbf{K}\mathbf{R}(t)\mathbf{K}^{-1}$.

Le problème *PRN2* peut également être résolu en utilisant le principe du minimum de Pontryagin et fournit la solution au problème initial. Les solutions aux problèmes *PR1* et *PR2*, ainsi qu'une propriété très intéressante, sont données par la Proposition 1.

Proposition 1. La solution aux problèmes *PR1* et *PR2* est donnée par :

$$\mathbf{G}(t) \propto \mathbf{G}_0 \Gamma(\mathbf{g}_0, q(t)) \quad (5.6)$$

où $q(t) = t$ pour *PR1* et $q(t) = -2t^3 + 3t^2$ pour *PR2* et :

$$\Gamma(\mathbf{g}_0, q(t)) = \mathbf{I} + \frac{\sin(\|\mathbf{g}_0\|q(t))}{\|\mathbf{g}_0\|} \{\mathbf{g}_0\} + \frac{1 - \cos(\|\mathbf{g}_0\|q(t))}{\|\mathbf{g}_0\|^2} \{\mathbf{g}_0\}^2 \quad (5.7)$$

avec :

$$\theta = \|\mathbf{g}_0\| = \arccos\left(\frac{1}{2}(D_0^{-1/3}T_0 - 1)\right) \quad (5.8)$$

$$\{\mathbf{g}_0\} = \frac{\|\mathbf{g}_0\|}{2 \sin \|\mathbf{g}_0\|} \left(D_0^{1/3} \mathbf{G}_0^+ - D_0^{-1/3} \mathbf{G}_0 \right) \quad (5.9)$$

où $D_0 = \det(\mathbf{G}_0)$, $T_0 = \text{Trace}(\mathbf{G}_0)$.

La solution obtenue est indépendante de la matrice \mathbf{K} , c'est-à-dire de la calibration de la caméra.

Preuve. La trajectoire de la matrice de rotation dans $SO(3)$ correspondant à la solution aux problèmes *PRN1* et *PRN2* peut être obtenue en utilisant la condition nécessaire du premier ordre du problème de contrôle optimal. La preuve est similaire à celle donnée en annexe B dans le cas d'un mouvement rigide. Une autre méthode est donnée dans [Zefran 96] pour obtenir la solution :

$$\mathbf{R}(t) = \mathbf{R}_0 e^{[\boldsymbol{\theta}_0]_{\times} q(t)} \quad (5.10)$$

où $[\boldsymbol{\theta}_0]_{\times} = \log(\mathbf{R}_0^T)$, $q(t) = t$ pour *PR1* et $q(t) = -2t^3 + 3t^2$ pour *PR2*. Dans le cas d'un mouvement de rotation pure, on obtient aisément (en utilisant (5.1), i.e $\mathbf{G} = \alpha \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$ avec $\alpha = \det(\mathbf{G})^{1/3}$) :

$$\begin{cases} \mathbf{K}\mathbf{R}\mathbf{K}^{-1} = D^{-1/3} \mathbf{G} \\ \mathbf{K}\mathbf{R}^T\mathbf{K}^{-1} = D^{1/3} \mathbf{G}^{-1} \\ \text{Trace}(\mathbf{R}) = D^{-1/3} T \end{cases} \quad (5.11)$$

avec $D = \det(\mathbf{G})$ et $T = \text{Trace}(\mathbf{G})$. En injectant (5.10) dans (5.1), on aboutit à :

$$\mathbf{G}(t) \propto \mathbf{K}\mathbf{R}_0\mathbf{e}^{[\boldsymbol{\theta}_0] \times q(t)}\mathbf{K}^{-1} = \mathbf{K}\mathbf{R}_0\mathbf{K}^{-1}\mathbf{K}\mathbf{e}^{[\boldsymbol{\theta}_0] \times q(t)}\mathbf{K}^{-1}$$

Cette dernière équation peut se mettre sous une forme plus compacte :

$$\mathbf{G}(t) \propto \mathbf{G}_0\mathbf{K}\mathbf{e}^{[\boldsymbol{\theta}_0] \times q(t)}\mathbf{K}^{-1} = \mathbf{G}_0\boldsymbol{\Gamma} \quad (5.12)$$

où $\boldsymbol{\Gamma} = \mathbf{K}\mathbf{e}^{[\boldsymbol{\theta}_0] \times q(t)}\mathbf{K}^{-1}$ et d'après (5.2) :

$$\boldsymbol{\Gamma} = \mathbf{I} + \frac{\sin \|\boldsymbol{\theta}_0\|}{\|\boldsymbol{\theta}_0\|}\mathbf{K}[\boldsymbol{\theta}_0]_{\times}\mathbf{K}^{-1} + \frac{1 - \cos \|\boldsymbol{\theta}_0\|}{\|\boldsymbol{\theta}_0\|^2}\mathbf{K}[\boldsymbol{\theta}_0]_{\times}^2\mathbf{K}^{-1}$$

Sur la base de l'équation précédente, on obtient :

$$\boldsymbol{\Gamma}(\mathbf{g}_0, q(t)) = \mathbf{I} + \frac{\sin(\|\mathbf{g}_0\|q(t))}{\|\mathbf{g}_0\|}\{\mathbf{g}_0\} + \frac{1 - \cos(\|\mathbf{g}_0\|q(t))}{\|\mathbf{g}_0\|^2}\{\mathbf{g}_0\}^2 \quad (5.13)$$

avec $\|\mathbf{g}_0\| = \|\boldsymbol{\theta}_0\|$ et $\{\mathbf{g}_0\} = \mathbf{K}[\boldsymbol{\theta}_0]_{\times}\mathbf{K}^{-1}$. En outre, nous avons d'après (5.3) :

$$\{\mathbf{g}_0\} = \mathbf{K} \log(\mathbf{R}_0^T)\mathbf{K}^{-1} = \frac{\theta}{2 \sin \theta}(\mathbf{K}\mathbf{R}_0^T\mathbf{K}^{-1} - \mathbf{K}\mathbf{R}_0\mathbf{K}^{-1})$$

En introduisant (5.11) dans la relation précédente on obtient :

$$\{\mathbf{g}_0\} = \frac{\theta}{2 \sin \theta} \left(D_0^{1/3}\mathbf{G}_0^{-1} - D_0^{-1/3}\mathbf{G}_0 \right) \quad (5.14)$$

et finalement, en utilisant (5.11) et (5.4), nous avons :

$$\|\boldsymbol{\theta}_0\| = \|\mathbf{g}_0\| = \arccos \left(\frac{1}{2}(D_0^{-1/3}T_0 - 1) \right). \quad (5.15)$$

Donc, d'après (5.12), (5.13), (5.14) et (5.15), la trajectoire de la colinéation correspondant aux problèmes *PR1* et *PR2* est donnée par la proposition 1.

Les termes $\{\mathbf{g}_0\}$ et $\|\mathbf{g}_0\|$ intervenant dans la proposition 1 sont directement obtenus à partir de la matrice de colinéation initiale \mathbf{G}_0 . En outre, \mathbf{G}_0 est estimée uniquement à partir de primitives géométriques extraites de l'image exprimées en pixel. Par conséquent, la trajectoire optimale de la matrice de colinéation est indépendante des paramètres intrinsèques de la caméra.

Remarque. Comme attendu, les trajectoires de la caméra sont dans les deux cas des géodésiques et ont une géométrie identique. Seules leurs paramétrisations temporelles diffèrent.

5.1.2 Mouvement de translation pure

Nous étudions dans cette partie le cas où la caméra effectue un mouvement de translation. La matrice de colinéation a alors la forme particulière suivante (se référer aux équations (2.13) et (2.20)):

$$\mathbf{G} \propto \mathbf{I} + \mathbf{K} \mathbf{t}_{df} \mathbf{n}^{fT} \mathbf{K}^{-1} \quad (5.16)$$

où $\mathbf{t}_{df} = \mathbf{t}/d^f$. Comme dans le cas d'une rotation pure, nous cherchons à obtenir les solutions des problèmes à énergie minimale (problème *PT1*) et à accélération minimale (problème *PT2*).

Problème PT1. On cherche dans ce cas la trajectoire de la matrice de colinéation qui correspond au problème à minimum d'énergie :

Trouver $\mathbf{G}(t)$ minimisant $J_3 = \int_0^1 \mathbf{v}^T \mathbf{v} dt$
avec :

$$\mathbf{v} = \dot{\mathbf{t}}$$

$$\mathbf{G} \propto \mathbf{I} + \mathbf{K} \mathbf{t}_{df} \mathbf{n}^{fT} \mathbf{K}^{-1}$$

et les conditions aux limites suivantes :

$$\mathbf{G}(0) \propto \mathbf{G}_0,$$

$$\mathbf{G}(1) \propto \mathbf{I}$$

Les conditions aux limites du problème *PT1* sont satisfaites pour les valeurs initiale et finale suivantes du vecteur de translation :

$$\mathbf{t}_{df}(0) = \mathbf{t}_{df0} \quad \text{et} \quad \mathbf{t}_{df}(1) = \mathbf{0}$$

Lorsqu'on substituera ces nouvelles conditions aux limites aux conditions initiales, on renommera *PTN1* le problème *PT1*. Le problème *PTN1* peut être scindé en deux parties comme pour le problème *PR1*. On cherche dans un premier temps la trajectoire du vecteur de translation partielle vérifiant les conditions aux limites et minimisant J_3 . On en déduit ensuite la matrice de colinéation correspondante. Cette solution est alors également la solution du problème initial *PT1*.

Problème PT2 : On cherche ici la solution correspondant au problème à minimum d'accélération qui s'écrit de la manière suivante :

Trouver $\mathbf{G}(t)$ minimisant $J_4 = \int_0^1 \dot{\mathbf{v}}^T \dot{\mathbf{v}} dt$
avec :

$$\mathbf{v} = \dot{\mathbf{t}}$$

$$\mathbf{G} \propto \mathbf{I} + \mathbf{K} \mathbf{t}_{df} \mathbf{n}^T \mathbf{K}^{-1}$$

et les conditions aux limites suivantes :

$$\mathbf{G}(0) \propto \mathbf{G}_0,$$

$$\mathbf{G}(1) \propto \mathbf{I}$$

$$\mathbf{v}(0) = \mathbf{v}(1) = \mathbf{0}$$

On définit le problème *PTN2* en substituant les conditions aux limites suivantes :

$$\mathbf{t}_{df}(0) = \mathbf{t}_{df0}; \quad \mathbf{t}_{df}(1) = \mathbf{0}; \quad \mathbf{v}(0) = \mathbf{v}(1) = \mathbf{0}.$$

à celles du problème initial *PT2*. La solution du problème *PTN2* est également solution du problème initial *PT2*.

Les solutions des problèmes *PT1* et *PT2* et le comportement vis-à-vis d'éventuelles erreurs sur les paramètres intrinsèques sont donnés par la Proposition 2.

Proposition 2. La trajectoire optimale de la matrice de colinéation au sens défini par *PT1* et *PT2* est donnée par :

$$\mathbf{G}(t) \propto q(t)\mathbf{I} + \frac{(1 - q(t))}{\alpha_0} \mathbf{G}_0 \quad (5.17)$$

où $q(t) = t$ pour *PT1* et $q(t) = -2t^3 + 3t^2$ pour *PT2* et où α_0 est une solution réelle de l'équation polynômiale de degré 3 suivante :

$$2\alpha^3 - T_0\alpha^2 + D_0 = 0 \quad (5.18)$$

La solution obtenue est indépendante de la matrice \mathbf{K} , c'est-à-dire de la calibration de la caméra.

Preuve. La trajectoire de la translation partielle correspondant aux problèmes *PTN1* et *PTN2* est donnée par (voir annexe B):

$$\mathbf{t}_{df}(t) = (1 - q(t))\mathbf{t}_{df0} \quad (5.19)$$

où $q(t) = t$ pour *PTN1* et $q(t) = -2t^3 + 3t^2$ pour *PTN2*. En introduisant (5.19) dans (5.16), nous obtenons :

$$\begin{aligned} \mathbf{G}(t) &\propto \mathbf{I} + \mathbf{K}(1 - q(t))\mathbf{t}_{df0}\mathbf{n}^{fT}\mathbf{K}^{-1} \\ &\propto q(t)\mathbf{I} + (1 - q(t))(\mathbf{I} + \mathbf{K}\mathbf{t}_{df0}\mathbf{n}^{fT}\mathbf{K}^{-1}) \end{aligned}$$

Nous déduisons (5.17) en notant que :

$$\mathbf{G}_0 = \alpha_0(\mathbf{I} + \mathbf{K}\mathbf{t}_{df0}\mathbf{n}^{fT}\mathbf{K}^{-1}) \quad (5.20)$$

où α_0 est un réel non nul. A partir de (5.20), nous obtenons les relations suivantes :

$$\begin{cases} \text{Trace}(G_0) = T_0 = \alpha_0(3 + \mathbf{t}_{df0}^T\mathbf{n}^f) \\ \text{Det}(G_0) = D_0 = \alpha_0^3(1 + \mathbf{t}_{df0}^T\mathbf{n}^f) \end{cases}$$

et en combinant les deux relations précédentes, nous déduisons que α_0 est solution de l'équation (5.18).

En outre, la valeur initiale de la matrice de colinéation \mathbf{G}_0 est directement extraite de primitives visuelles exprimées en pixel et α_0 peut directement être obtenue à partir de \mathbf{G}_0 (voir (5.18)). La matrice de colinéation à l'instant t ne dépendant que de \mathbf{G}_0 et α_0 , $\mathbf{G}(t)$ est donc indépendante de la calibration de la caméra.

Remarque. La trajectoire de la colinéation obtenue correspond dans les deux cas à une trajectoire minimale de la caméra, c'est-à-dire ici à un mouvement rectiligne du centre de projection de la caméra. Une nouvelle fois, seule l'évolution temporelle diffère en fonction de la fonction de coût minimisée.

5.1.3 Mouvement rigide

Nous supposons maintenant que le repère attaché à la caméra à l'instant t , \mathcal{F}_c , subit la transformation :

$$\mathbf{D}(t) = \begin{bmatrix} \mathbf{R}(t) & \mathbf{t}(t) \\ \mathbf{0} & 1 \end{bmatrix}$$

de telle sorte que celui-ci se déplace du repère initiale \mathcal{F}_i à l'instant 0 jusqu'au repère caméra final \mathcal{F}_f à l'instant $t = 1$. En notant \mathbf{U} le vecteur $[\mathbf{v}^T\boldsymbol{\omega}^T]^T$ de dimension 6×1 , les problèmes à minimum d'énergie (*problème PC1*) et à minimum d'accélération (*problème PC2*) peuvent se mettre sous forme standard. Nous rappelons que dans ce cas la matrice de colinéation relative à un plan de référence s'écrit :

$$\mathbf{G}(t) \propto \mathbf{K}(\mathbf{R} + \mathbf{t}_{df}\mathbf{n}^{fT})\mathbf{K}^{-1} \quad (5.21)$$

Problème PC1. On cherche dans ce cas la trajectoire de la matrice de colinéation qui correspond au problème à minimum d'énergie :

Trouver $\mathbf{G}(t)$ minimisant $J_5 = \int_0^1 \mathbf{U}^T \mathbf{U} dt$
avec :

$$\mathbf{R}^T \dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times},$$

$$\mathbf{v} = \dot{\mathbf{t}}$$

$$\mathbf{G}(t) \propto \mathbf{K} \left(\mathbf{R} + \mathbf{t}_{df} \mathbf{n}^{fT} \right) \mathbf{K}^{-1}$$

et les conditions aux limites suivantes :

$$\mathbf{G}(0) \propto \mathbf{G}_0$$

$$\mathbf{G}(1) \propto \mathbf{I}$$

Les conditions aux limites du problème *PC1* sont satisfaites si $\mathbf{R}(0) = \mathbf{R}_0$, $\mathbf{t}_{df}(0) = \mathbf{t}_{df0}$, $\mathbf{R}(1) = \mathbf{I}$ et $\mathbf{t}_{df}(1) = \mathbf{0}$. Ce nouveau problème sera noté *PCN1*, les solutions de *PCN1* sont également solutions de *PC1*.

Problème PC2. Nous cherchons ici la trajectoire de la matrice de colinéation qui correspond au problème à minimum d'accélération :

Trouver $\mathbf{G}(t)$ minimisant $J_6 = \int_0^1 \dot{\mathbf{U}}^T \dot{\mathbf{U}} dt$
avec :

$$\mathbf{R}^T \dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times},$$

$$\mathbf{v} = \dot{\mathbf{t}}$$

$$\mathbf{G}(t) \propto \mathbf{K} \left(\mathbf{R} + \mathbf{t}_{df} \mathbf{n}^{fT} \right) \mathbf{K}^{-1}$$

et les conditions aux limites suivantes :

$$\mathbf{G}(0) \propto \mathbf{G}_0$$

$$\mathbf{G}(1) \propto \mathbf{I}$$

$$\mathbf{U}(0) = \mathbf{U}(1) = \mathbf{0}$$

Une nouvelle fois, notons que les conditions aux limites sont satisfaites si $\mathbf{R}(0) = \mathbf{R}_0$, $\mathbf{t}_{df}(0) = \mathbf{t}_{df0}$, $\mathbf{R}(1) = \mathbf{I}$, $\mathbf{t}_{df}(1) = \mathbf{0}$ et $\mathbf{U}(0) = \mathbf{U}(1) = \mathbf{0}$ (problème *PCN2*). Les solutions des problèmes *PC1* et *PC2* et la dépendance vis-à-vis de la calibration de la caméra sont données par la Proposition 3.

Proposition 3. La trajectoire optimale de la matrice de colinéation au sens défini par *PC1* et *PC2* est donnée par :

$$\mathbf{G}(t) \propto (1 - q(t))\Phi_0 + (\mathbf{G}_0 - \Phi_0)\Gamma(\boldsymbol{\theta}_0, t) \quad (5.22)$$

où $q(t) = t$ pour *PT1* et $q(t) = -2t^3 + 3t^2$ pour *PT2* et :

$$\begin{cases} \Gamma(\boldsymbol{\theta}_0, q(t)) = \mathbf{K}e^{[\boldsymbol{\theta}_0]_{\times} q(t)} \mathbf{K}^{-1} \\ \Phi_0 = \mathbf{K} \mathbf{t}_{df0} \mathbf{n}^{fT} \mathbf{K}^{-1} \end{cases} \quad (5.23)$$

avec $[\boldsymbol{\theta}_0]_{\times} = \log(\mathbf{R}_0^T)$

Les solutions aux problèmes PC1 et PC2 sont indépendantes de la calibration de la caméra si $\mathbf{G}_{0\infty}$ est connue.

Preuve. Les trajectoires de la translation et de la rotation correspondant aux problèmes *PCN1* et *PCN2* sont données par (voir l'annexe B pour la démonstration) :

$$\begin{cases} \mathbf{t}_{df}(t) = (1 - q(t))\mathbf{t}_{df0} \\ \mathbf{R}(t) = \mathbf{R}_0 e^{[\boldsymbol{\theta}_0]_{\times} q(t)} \end{cases}$$

En introduisant ces dernières relations dans (5.21), la trajectoire de la colinéation peut s'écrire sous la forme suivante :

$$\mathbf{G}(t) \propto \mathbf{K}(\mathbf{R}_0 e^{[\boldsymbol{\theta}_0]_{\times} q(t)} + (1 - q(t))\mathbf{t}_{df0} \mathbf{n}^{fT}) \mathbf{K}^{-1}$$

L'équation précédente peut se réécrire :

$$\mathbf{G}(t) \propto \mathbf{K} \left(\mathbf{R}_0 e^{[\boldsymbol{\theta}_0]_{\times} q(t)} + (1 - q(t))\mathbf{t}_{df0} \mathbf{n}^{fT} + \mathbf{t}_{df0} \mathbf{n}^{fT} e^{[\boldsymbol{\theta}_0]_{\times} q(t)} - \mathbf{t}_{df0} \mathbf{n}^{fT} e^{[\boldsymbol{\theta}_0]_{\times} q(t)} \right) \mathbf{K}^{-1}$$

et se mettre sous la forme donnée par (5.22) en posant :

$$\begin{cases} \Gamma(\boldsymbol{\theta}_0, q(t)) = \mathbf{K}e^{[\boldsymbol{\theta}_0]_{\times} q(t)} \mathbf{K}^{-1} \\ \Phi_0 = \mathbf{K} \mathbf{t}_{df0} \mathbf{n}^{fT} \mathbf{K}^{-1} \end{cases}$$

On remarque que la matrice contenant les paramètres intrinsèques de la caméra apparaît explicitement dans l'expression de la trajectoire donnée par (5.22). Cependant, nous allons voir que lorsqu'on utilise la matrice non singulière, $\hat{\mathbf{K}}$, contenant

des paramètres intrinsèques erronés, à la place de la vraie matrice \mathbf{K} , la trajectoire de la colinéation n'est pas affectée. En effet, la matrice de colinéation initiale \mathbf{G}_0 n'est pas affectée par d'éventuelles erreurs sur \mathbf{K} puisqu'elle est directement obtenue à partir des données en pixel extraites des images initiale et désirée. En outre nous avons :

$$\begin{cases} \widehat{\Phi}_0 = \widehat{\mathbf{K}} \widehat{\mathbf{t}}_{0df} \widehat{\mathbf{n}}^{fT} \widehat{\mathbf{K}}^{-1} \\ \widehat{\Gamma}(\boldsymbol{\theta}_0, q(t)) = \widehat{\mathbf{K}} \mathbf{e}^{[\widehat{\boldsymbol{\theta}}_0]_{\times} q(t)} \widehat{\mathbf{K}}^{-1} \end{cases} \quad (5.24)$$

En introduisant (2.40) et (2.41) dans $\widehat{\Phi}_0$ (ces relations sont strictement exactes si $\widehat{\mathbf{G}}_{0\infty}$ est disponible), on obtient :

$$\begin{aligned} \widehat{\Phi}_0 &= \widehat{\mathbf{K}} \|\mathbf{n}^{fT} \delta \mathbf{K}^{-1}\| \delta \mathbf{K} \mathbf{t}_{df} \frac{\mathbf{n}^{fT} \delta \mathbf{K}^{-1}}{\|\mathbf{n}^{fT} \delta \mathbf{K}^{-1}\|} \mathbf{K}^{-1} \\ &= \mathbf{K} \mathbf{t}_{df0} \mathbf{n}^{fT} \mathbf{K}^{-1} = \Phi_0 \end{aligned}$$

et d'après (5.3) :

$$[\widehat{\boldsymbol{\theta}}_0]_{\times} = \log(\mathbf{R}_0^T) = \frac{\widehat{\theta}}{2 \sin(\widehat{\theta})} (\widehat{\mathbf{R}}_0^T - \widehat{\mathbf{R}}_0) = \widehat{\kappa} (\widehat{\mathbf{R}}_0^T - \widehat{\mathbf{R}}_0) \quad (5.25)$$

De plus, $\widehat{\theta} = \theta$ (se référer à l'équation (2.42)) implique $\widehat{\kappa} = \kappa$. En combinant les relations (5.25) et (5.2), nous obtenons :

$$\widehat{\Gamma}(\boldsymbol{\theta}_0, q(t)) = \widehat{\mathbf{K}} \left(\mathbf{I} + \kappa \frac{\sin(\|\boldsymbol{\theta}_0\| q(t))}{\|\boldsymbol{\theta}_0\|} (\widehat{\mathbf{R}}_0^T - \widehat{\mathbf{R}}_0) + \kappa^2 \frac{1 - \cos(\|\boldsymbol{\theta}_0\| q(t))}{\|\boldsymbol{\theta}_0\|^2} (\widehat{\mathbf{R}}_0^T - \widehat{\mathbf{R}}_0)^2 \right) \widehat{\mathbf{K}}^{-1}$$

En notant que $\widehat{\mathbf{K}} \widehat{\mathbf{R}}_0 \widehat{\mathbf{K}}^{-1} = \mathbf{K} \mathbf{R}_0 \mathbf{K}^{-1}$, nous aboutissons finalement à :

$$\widehat{\Gamma}(\widehat{\boldsymbol{\theta}}_0, q(t)) = \Gamma(\boldsymbol{\theta}_0, q(t))$$

Donc, la trajectoire de la matrice de colinéation donnée par (5.22) est indépendante du choix de la matrice non singulière $\widehat{\mathbf{K}}$ si $\widehat{\mathbf{G}}_{0\infty}$ est connue.

Remarque. Notons que le second terme de l'équation (5.22) correspond en fait à la matrice d'homographie à l'infini à l'instant t dans le domaine pixelique :

$$(\mathbf{G}_0 - \Phi_0) \Gamma = \mathbf{K} \mathbf{H}_{\infty} \mathbf{K}^{-1} \quad (5.26)$$

Notons également que les conditions aux limites des problèmes *PR2*, *PT2* et *PC2* sont effectivement les conditions aux limites souhaitées dans le contexte de l'asservissement visuel et que les résultats obtenus correspondent à des mouvements de la caméra le long d'une géodésique.

Dans la partie suivante, nous utilisons la trajectoire de la matrice de colinéation relative à un plan de référence afin d'obtenir une forme analytique de la trajectoire des pixels issus de la projection d'une scène plane ou non plane dans l'image.

5.1.4 Les trajectoires dans l'image

Notons $\mathbf{p}(t) = [u(t) \ v(t) \ 1]^T$ les coordonnées homogènes d'un pixel dans l'image à l'instant t , et \mathbf{p}^f son correspondant dans l'image finale (se référer à la figure 5.1). Comme nous l'avons déjà vu, on a :

$$\mathbf{h}(t) = \alpha(t)\mathbf{p}(t) = \mathbf{G}(t)\mathbf{p}^f + \beta\mathbf{e}(t) \quad (5.27)$$

où $\mathbf{h}(t)$ est un vecteur de dimension 3 que nous introduisons pour simplifier les notations, $\alpha(t)$ un réel dépendant du temps et β un scalaire constant nul si le point $3D$, \mathcal{M} , se projetant en $\mathbf{p}(t)$ dans l'image à l'instant t appartient au plan Π de référence. Nous rappelons également que le vecteur $\mathbf{e}(t) \propto \mathbf{K}\mathbf{t}(t)$ représente l'épipole. Les coordonnées de l'épipole à l'instant $t = 0$, $\mathbf{e}(0) = \mathbf{e}_0$, peuvent directement être obtenues à partir des primitives extraites de l'image (i.e l'estimation de \mathbf{e}_0 n'est pas affectée par d'éventuelles erreurs sur la calibration de la caméra). En outre, on voit aisément à partir de la relation (5.19) que sa trajectoire correspondant au problème du minimum de vitesse ou d'accélération est donnée par :

$$\mathbf{e}(t) \propto (1 - q(t))\mathbf{e}_0 \quad (5.28)$$

où $q(t) = t$ pour les problèmes à minimum d'énergie, et $q(t) = -2t^3 + 3t^2$ pour les problèmes à minimum d'accélération. Clairement, une telle trajectoire de l'épipole n'est pas affectée par des erreurs sur les paramètres intrinsèques de la caméra. Notons également que le facteur β ne dépend pas du temps et donc peut être calculé directement à partir des données extraites des images initiale et finale¹ :

$$\beta = \text{signe} \left(-\frac{(\mathbf{G}_0\mathbf{p}^f \wedge \mathbf{p}(0))_1}{(\mathbf{e}_0 \wedge \mathbf{p}(0))_1} \right) \frac{\|\mathbf{G}_0\mathbf{p}^f \wedge \mathbf{p}(0)\|}{\|\mathbf{e}_0 \wedge \mathbf{p}(0)\|} \quad (5.29)$$

D'après la relation (5.27), le vecteur \mathbf{h} n'est pas affecté par des erreurs sur les paramètres intrinsèques car $\mathbf{G}(t)$, $\mathbf{e}(t)$ et β peuvent être obtenus sans erreur même si la matrice \mathbf{K} est mal estimée. Les trajectoires dans l'image des pixels correspondant à une trajectoire optimale de la caméra peuvent donc être calculées sans erreur, avec :

$$\begin{cases} u(t) = \frac{(\mathbf{h}(t))_1}{(\mathbf{h}(t))_3} \\ v(t) = \frac{(\mathbf{h}(t))_2}{(\mathbf{h}(t))_3} \end{cases} \quad (5.30)$$

La relation précédente peut par exemple être appliquée à des pixels correspondant à des points d'intérêt dans l'image, ou éventuellement à tous les pixels de l'image. Dans les deux parties suivantes, nous utilisons les résultats obtenus auparavant, d'une part, pour planifier la trajectoire de tous les pixels d'une image et, d'autre part, pour planifier la trajectoire de points d'intérêt extraits de l'image. Nous vérifions ensuite la robustesse de la planification vis-à-vis d'erreurs sur la calibration de la caméra.

1. $(\mathbf{x})_n$ désigne la $n^{\text{ième}}$ composante du vecteur \mathbf{x}

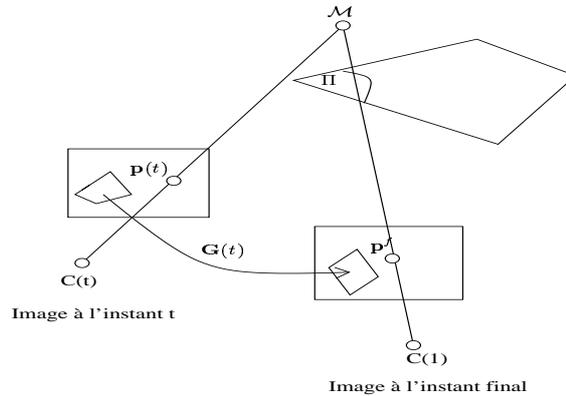


FIG. 5.1 – Image d'un pixel à deux instants

5.1.5 Algorithmes et résultats de planification

Les images initiale et désirée utilisées ont été acquises en utilisant la plate-forme robotique de l'IRISA. Nous avons réalisé six expérimentations, correspondant aux six cas de figure que nous avons étudiés :

- mouvement en rotation pure de la caméra et trajectoires à minimum d'énergie et à minimum d'accélération,
- mouvement en translation pure de la caméra et trajectoires à minimum d'énergie et à minimum d'accélération,
- mouvement rigide de la caméra et trajectoires à minimum d'énergie et à minimum d'accélération.

L'objectif de ces expérimentations est la planification de la trajectoire de tous les pixels de l'image.

5.1.5.1 Déplacement en rotation

Dans ces deux premières expériences, le déplacement de la caméra entre l'acquisition des images initiale et désirée (figures encadrées dans la représentation des trajectoires dans l'image) est une rotation pure. L'algorithme du schéma de planification est donné par la figure 5.2. La matrice de colinéation initiale est donnée par :

$$\mathbf{G}_0 = \begin{bmatrix} 0.3737 & 0.4865 & -232.5540 \\ -0.4248 & 0.3637 & -65.5569 \\ 0.0000 & 0.0001 & -0.5833 \end{bmatrix}$$

Des images intermédiaires, synthétisées en planifiant la trajectoire de tous les pixels de l'image dans les cas du minimum d'énergie et du minimum d'accélération, sont fournies par les figures 5.5 et 5.7 respectivement. Les trajectoires correspondantes de la caméra en fonction du temps sont illustrées sur les figures 5.6 et 5.8. Comme on peut le constater sur les figures 5.6 et 5.8, le comportement temporel est différent selon la fonction de coût utilisée.

1. Extraction des primitives visuelles
2. Mise en correspondances des primitives visuelles extraites
3. Obtention de \mathbf{G}_0 et \mathbf{e}_0 en utilisant [Malis 00]
4. Calcul de :
 - $\mathbf{\Gamma}(\mathbf{g}_0, q(t))$ (équation (5.7))
 - $\mathbf{G}(t)$ (équation (5.6))
 - $\mathbf{e}(t)$ (équation (5.28))
 - β (équation (5.29))
5. Obtention des primitives visuelles à l'instant t (équation (5.29))

FIG. 5.2 – *Algorithme dans le cas d'une pure rotation*

5.1.5.2 Déplacement en translation

Le mouvement de la caméra entre l'acquisition des images initiale et désirée est, cette fois, un mouvement de translation pure. L'algorithme de la planification est illustré sur la figure 5.3. La matrice de colinéation initiale est donnée par :

$$\mathbf{G}_0 = \begin{bmatrix} -0.6064 & 0.1277 & -359.2228 \\ -0.0291 & -0.4614 & -249.9619 \\ -0.0002 & 0.0005 & -1.3909 \end{bmatrix}$$

Les résultats sont fournis sur les figures 5.9, 5.10 et 5.11, 5.12 dans les cas du minimum d'énergie et du minimum d'accélération respectivement. Les trajectoires de la caméra sont maintenant représentées dans l'espace de travail du robot. La représentation de la caméra est accompagnée de l'instant correspondant.

5.1.5.3 Déplacement en rotation et translation

Dans le cas d'un mouvement en rotation et translation, nous avons considéré que la matrice de colinéation à l'infini était disponible pour montrer l'indépendance des trajectoires planifiées dans l'image vis-à-vis des paramètres intrinsèques. Notons une nouvelle fois que dans le cas d'une rotation pure ou d'une translation pure, cette matrice est toujours disponible. En pratique, nous utilisons l'algorithme proposé dans [Faugeras 88] pour obtenir les informations nécessaires à la constructions des trajectoires dans l'image pour un déplacement général (rotation, translation au facteur d'échelle et normale au plan de référence, voir figure 5.4 pour l'algorithme complet). Cet algorithme est conçu pour les cas où la calibration est connue et donc les trajectoires de la matrice de colinéation ne sont plus invariantes vis-à-vis de la calibration

1. Extraction des primitives visuelles
2. Mise en correspondances des primitives visuelles extraites
3. Obtention de \mathbf{G}_0 et \mathbf{e}_0 en utilisant [Malis 00]
4. Calcul de :
 - α_0 (équation (5.18))
 - $\mathbf{G}(t)$ (équation 5.17))
 - $\mathbf{e}(t)$ (équation 5.28))
 - β (équation (5.29))
5. Obtention des primitives visuelles à l'instant t (équation (5.29))

FIG. 5.3 – *Algorithme dans le cas d'une pure translation*

de la caméra. Son utilisation dans notre schéma de planification nécessite donc une étude de l'influence des paramètres intrinsèques. Dans le paragraphe suivant, nous verrons de manière expérimentale que notre schéma de planification est très robuste aux erreurs de calibration (quand [Faugeras 88] est utilisé). Cependant une étude théorique des relations entre les variations des trajectoires et les variations de la matrice des paramètres intrinsèques reste à effectuer. Cette étude passe par l'analyse de l'algorithme utilisé pour retrouver le déplacement de la caméra. Une analyse approfondie de celui-ci permettrait alors de répondre à de nombreuses questions restantes ouvertes dans le domaine de la vision pour la robotique. Par exemple, les schémas de commande utilisant une reconstruction projective puis euclidienne de la scène comme le 2D1/2 ont un domaine de convergence et de stabilité en présence d'erreurs sur les paramètres de la caméra qui est connu uniquement sous l'hypothèse que la matrice de colinéation à l'infini soit mesurable à chaque itération de la boucle de commande. Cette hypothèse pourrait être remplacée par la connaissance de l'influence des erreurs de calibration sur le schéma réel d'estimation du déplacement de la caméra (en général [Faugeras 88]).

Dans cette dernière série de résultats, le déplacement de la caméra entre l'acquisition des images initiale et désirée est un mouvement rigide. La matrice de colinéation initiale est donnée par :

$$\mathbf{G}_0 = \begin{bmatrix} 0.4603 & 0.4145 & -597.3284 \\ -0.2476 & 0.6107 & -376.0835 \\ -0.0001 & 0.0006 & -1.5281 \end{bmatrix}$$

Les résultats sont fournis sur les figures 5.13, 5.14 et 5.15, 5.16 dans les cas du minimum d'énergie et du minimum d'accélération respectivement. De la même manière

que précédemment, la trajectoire de la caméra est représentée dans l'espace de travail et chaque position de la caméra est accompagnée de l'instant correspondant.

1. Extraction des primitives visuelles
2. Mise en correspondances des primitives visuelles extraites
3. Obtention de \mathbf{G}_0 et \mathbf{e}_0 en utilisant [Malis 00]
4. Obtention de $\hat{\mathbf{H}}_0$ (équation (2.39))
5. Utilisation de [Faugeras 88] pour obtenir $\hat{\mathbf{R}}_0, \hat{\mathbf{t}}_{df0}, \hat{\mathbf{n}}^f$
6. Calcul de :
 - $\hat{\Phi}_0$ (équation (5.24))
 - $\hat{\Gamma}(\boldsymbol{\theta}_0, q(t))$ (équation (5.24))
 - $\mathbf{G}(t)$ (équation (5.22))
 - $\mathbf{e}(t)$ (équation (5.28))
 - β (équation (5.29))
7. Obtention des primitives visuelles à l'instant t (équation (5.29))

FIG. 5.4 – *Algorithme dans le cas d'un mouvement de rotation et translation*

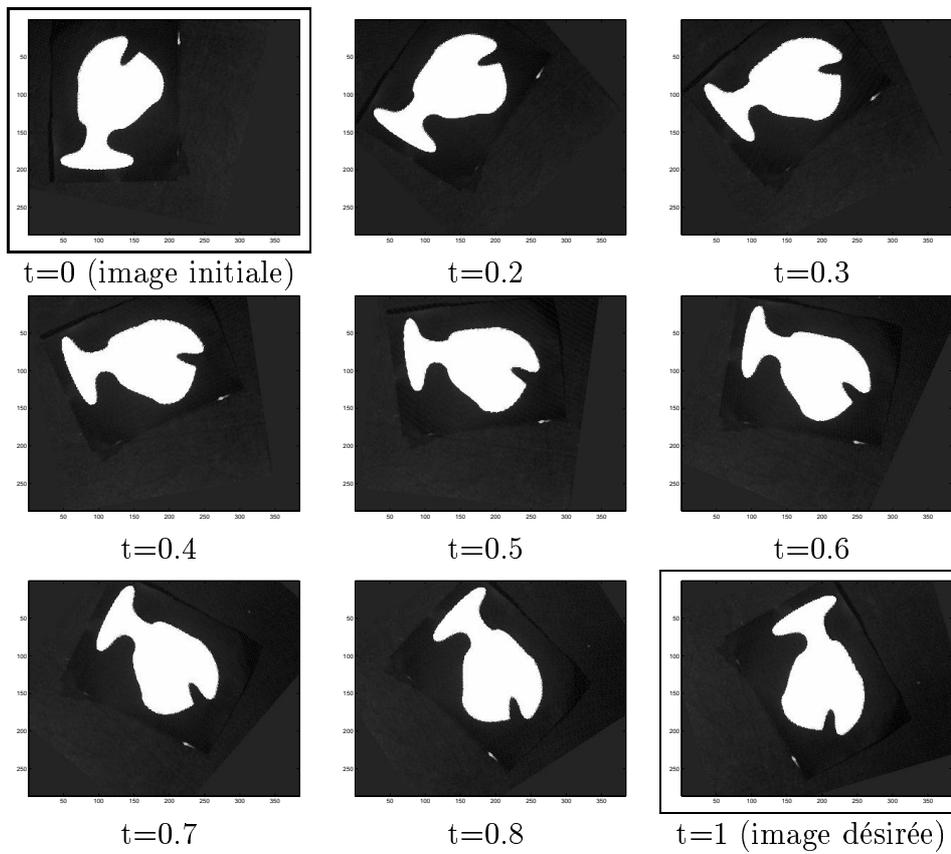


FIG. 5.5 – Trajectoire dans l'image: mouvement de rotation - minimum d'énergie

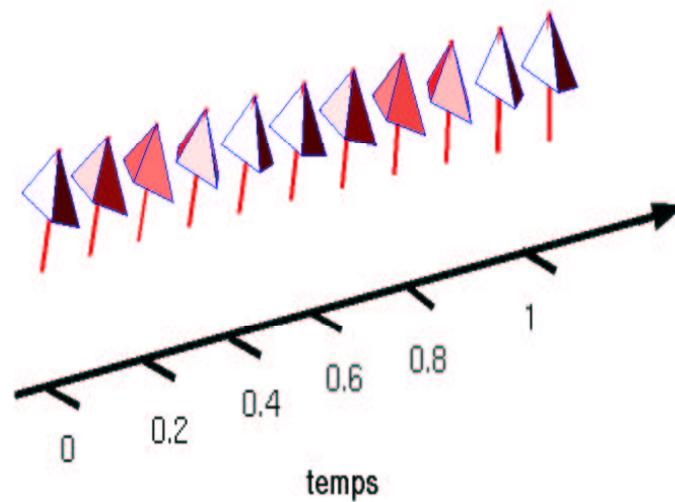


FIG. 5.6 – Trajectoire de la caméra: mouvement de rotation - minimum d'énergie

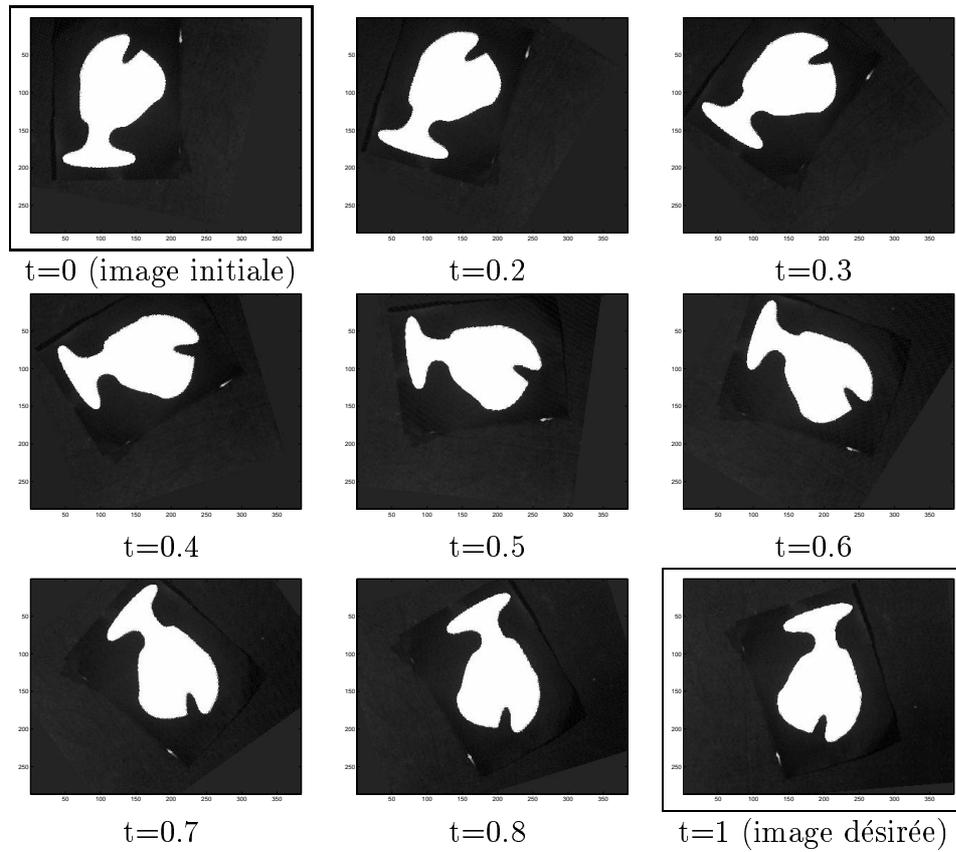


FIG. 5.7 – Trajectoire dans l'image: mouvement de rotation - minimum d'accélération

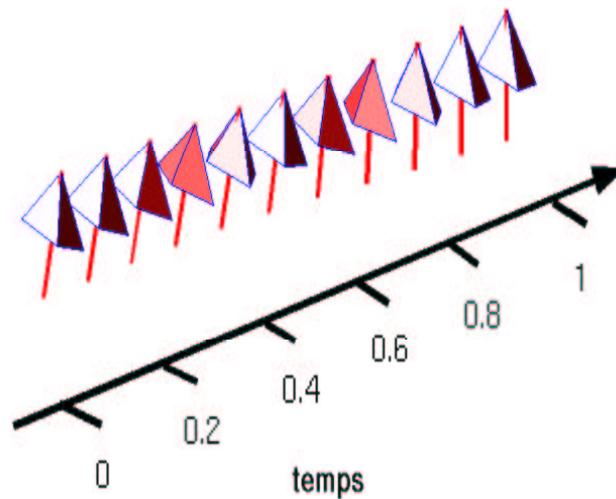


FIG. 5.8 – Trajectoire de la caméra: mouvement de rotation - minimum d'accélération

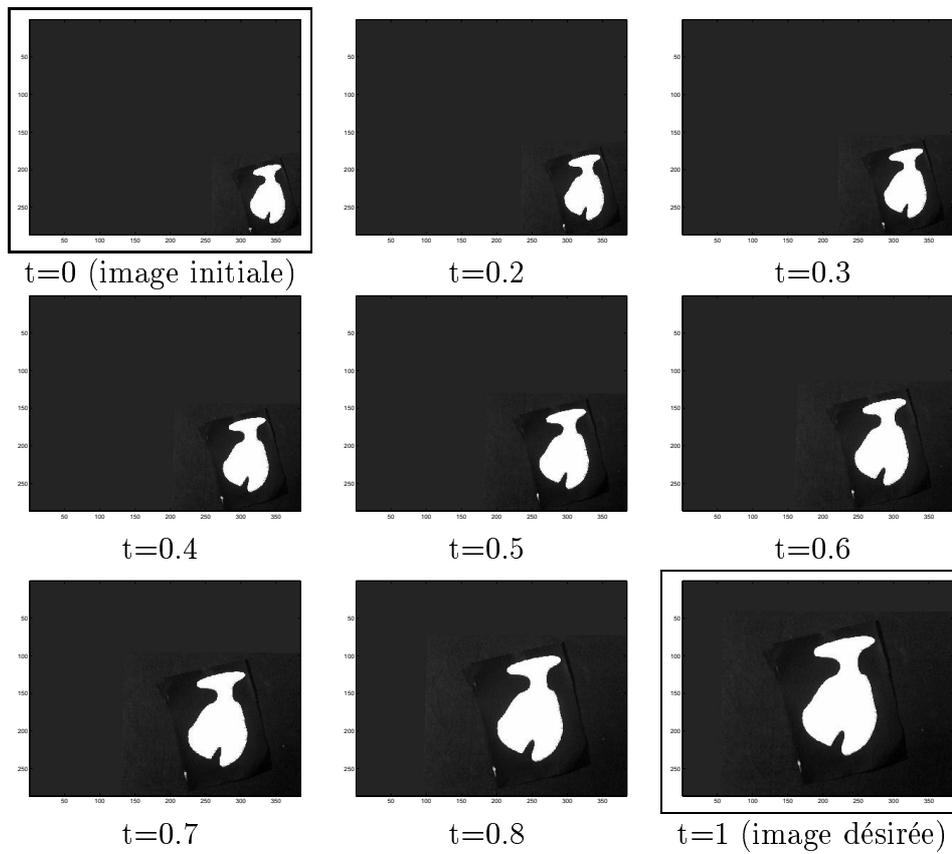


FIG. 5.9 – Trajectoire dans l'image : mouvement de translation - minimum d'énergie

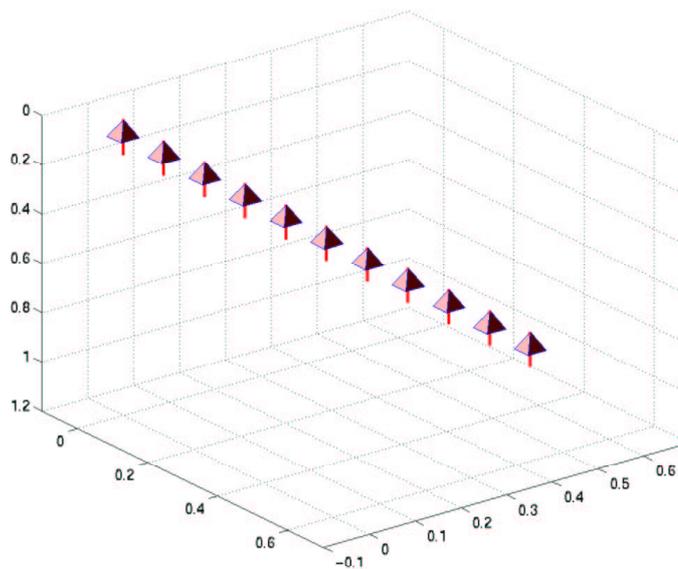


FIG. 5.10 – Trajectoire de la caméra : mouvement de translation - minimum d'énergie

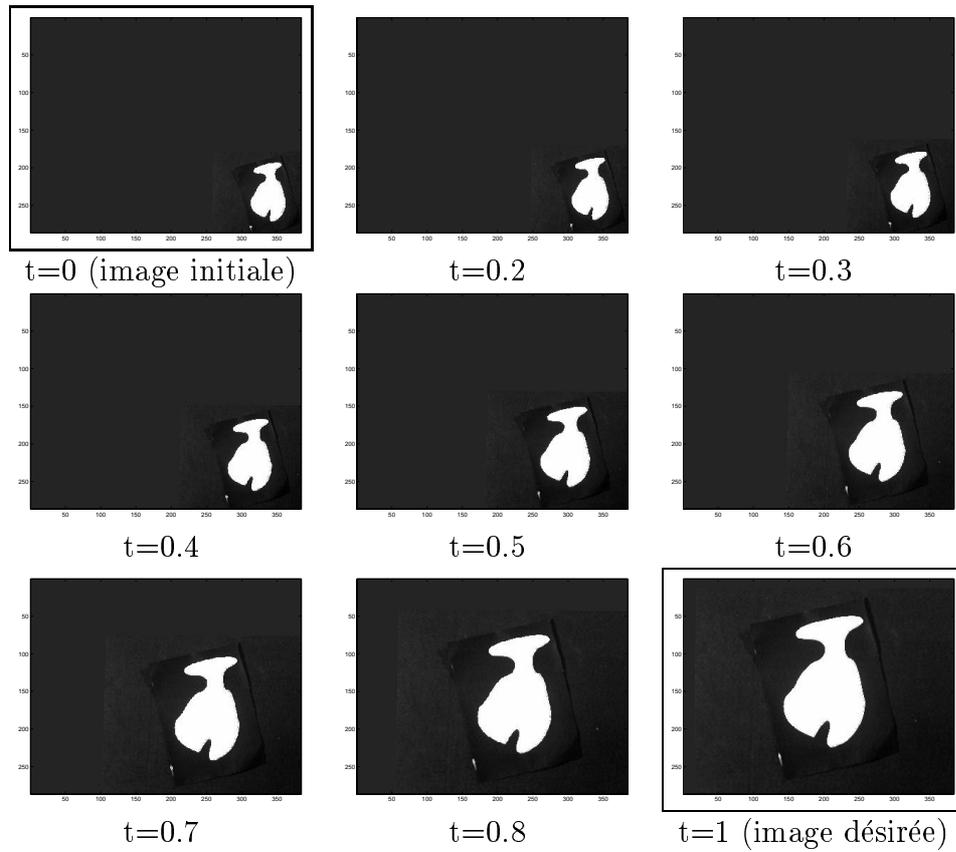


FIG. 5.11 – Trajectoire dans l'image: mouvement de translation - minimum d'accélération

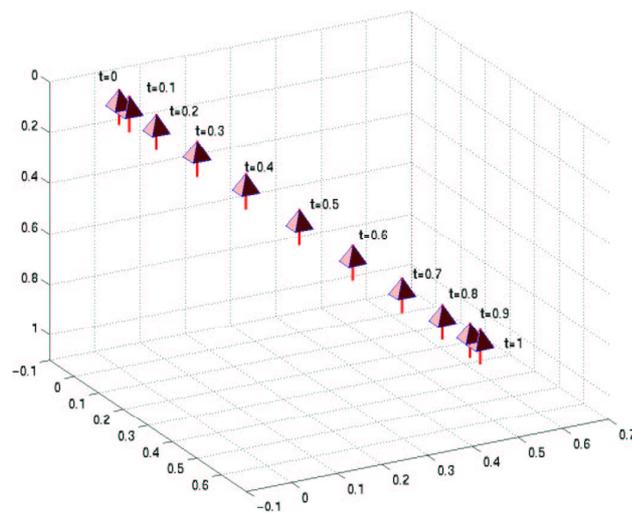


FIG. 5.12 – Trajectoire de la caméra: mouvement de translation - minimum d'accélération

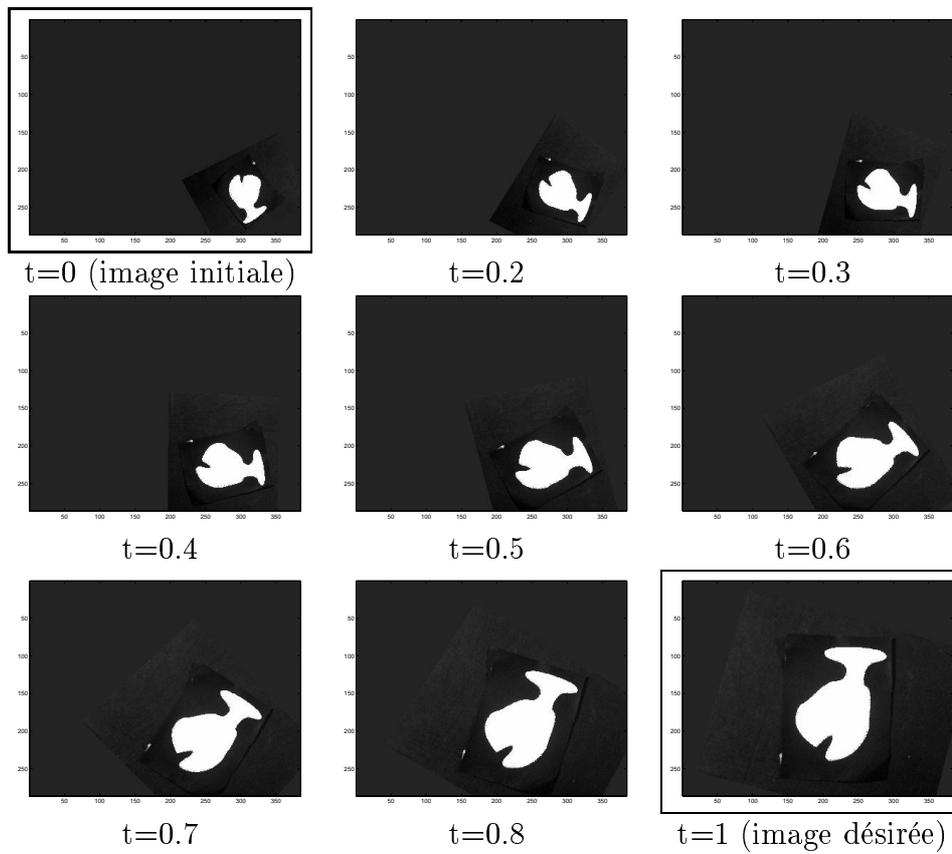


FIG. 5.13 – Trajectoire dans l'image: mouvement de translation et rotation - minimum d'énergie

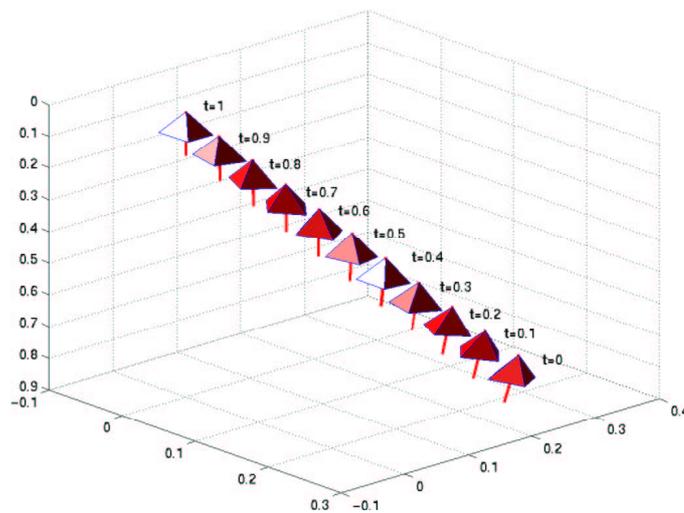


FIG. 5.14 – Trajectoire de la caméra: mouvement de translation et rotation - minimum d'énergie

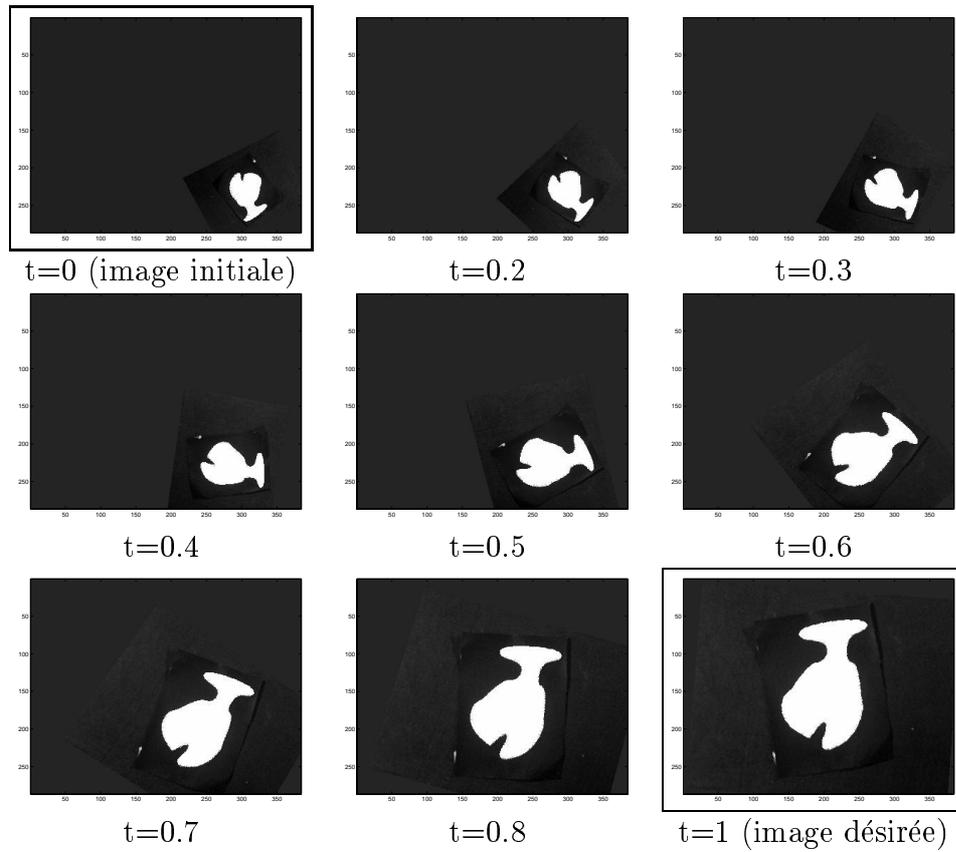


FIG. 5.15 – Trajectoire dans l'image: mouvement de translation et rotation - minimum d'accélération

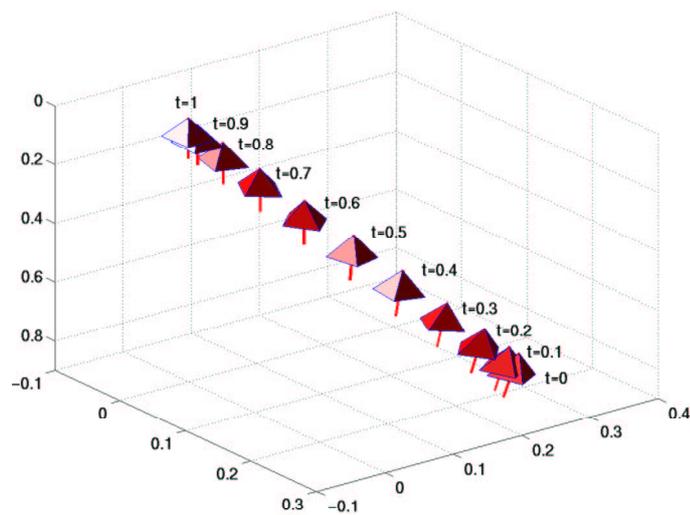


FIG. 5.16 – Trajectoire de la caméra: mouvement de translation et rotation - minimum d'accélération

5.1.5.4 Robustesse vis-à-vis des erreurs de calibration

En pratique, nous utilisons l'algorithme proposé dans [Faugeras 88] pour estimer les paramètres de mouvement à partir de la matrice d'homographie. Pour vérifier expérimentalement la robustesse de notre méthode vis-à-vis des paramètres de calibration lorsque nous utilisons cet algorithme, nous avons effectué un ensemble d'expériences en introduisant des erreurs sur la calibration de la caméra.

Nous réalisons la planification de trajectoires de points d'intérêt entre une image initiale et une image finale d'une scène complexe. Les points d'intérêt sont obtenus en utilisant le détecteur de Harris [Harris 88] et mis en correspondance en utilisant le logiciel image matching [Zhang 95a]. Les images correspondant aux positions initiale et désirée de la caméra sont données par la figure 5.17. Les points d'intérêt extraits et mis en correspondance dans les deux images sont indiqués par des carrés blancs.

La trajectoire dans l'image et celle de la caméra, obtenues en utilisant une calibration correcte de la caméra, sont illustrées sur les figures 5.18(a) et 5.18(b) respectivement. Les expérimentations suivantes ont été réalisées en introduisant des erreurs sur les paramètres de calibration :

- figure 5.19, une erreur de 20% a été ajoutée à chacun des paramètres intrinsèques,
- figure 5.20, une erreur de 40% a été ajoutée à chacun des paramètres intrinsèques,
- figure 5.21, une erreur de 60% a été ajoutée à chacun des paramètres intrinsèques,
- figure 5.22, une erreur de 80% a été ajoutée à chacun des paramètres intrinsèques.

En comparant les différents résultats, on observe que les trajectoires planifiées dans l'image sont peu modifiées lorsqu'on fait varier les paramètres intrinsèques. Les trajectoires de la caméra correspondantes sont également similaires. Cela confirme la robustesse de la planification de trajectoires vis-à-vis d'erreurs sur la calibration de caméra.

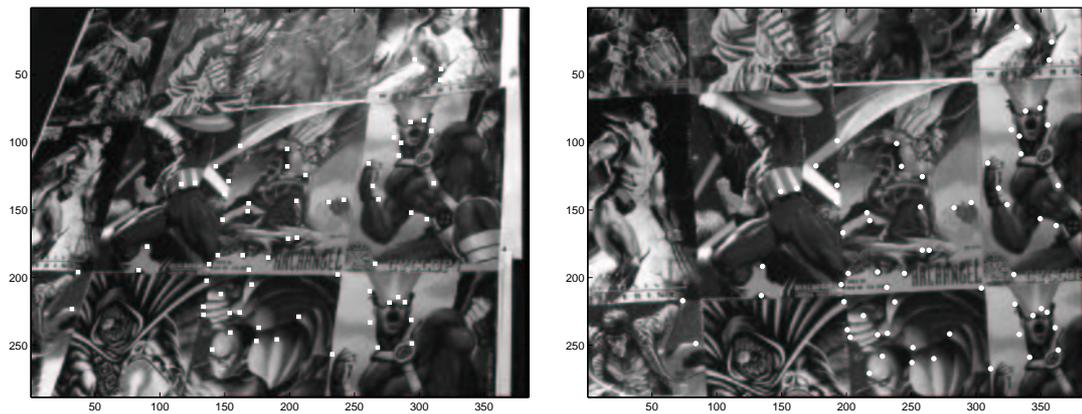
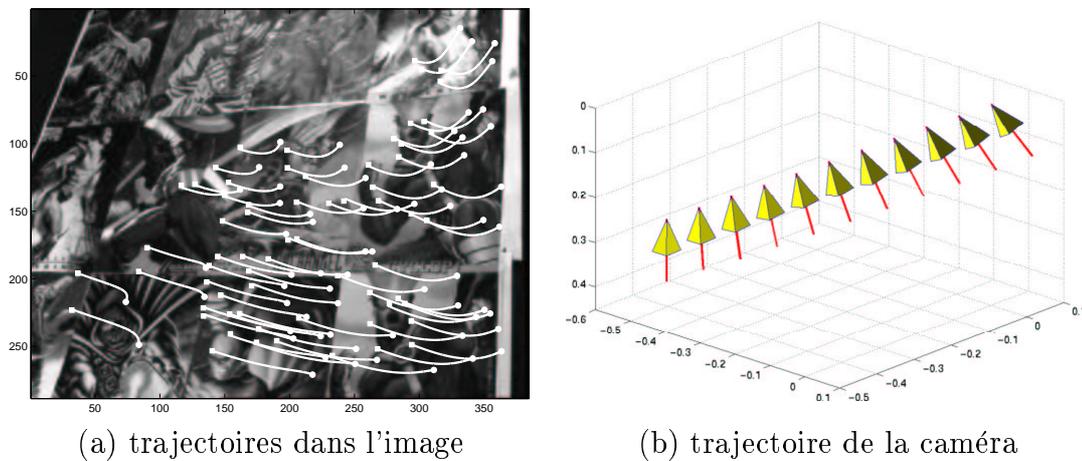


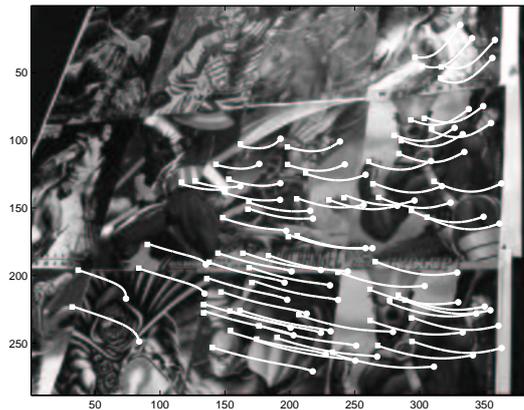
FIG. 5.17 – Image initiale (a) et image désirée (b)



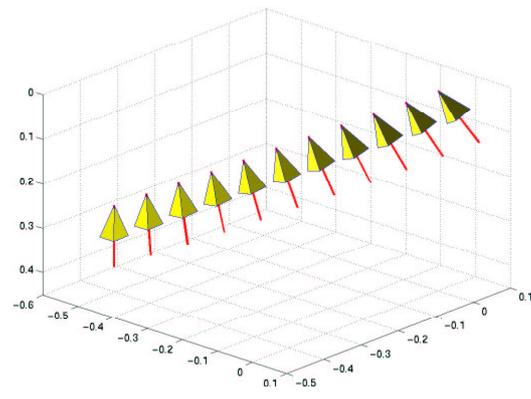
(a) trajectoires dans l'image

(b) trajectoire de la caméra

FIG. 5.18 – Paramètres de calibration corrects

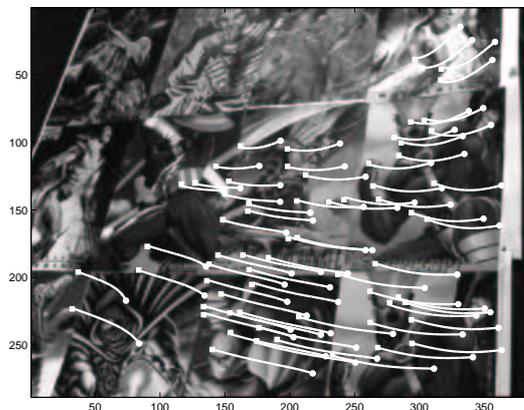


(a) trajectoires dans l'image

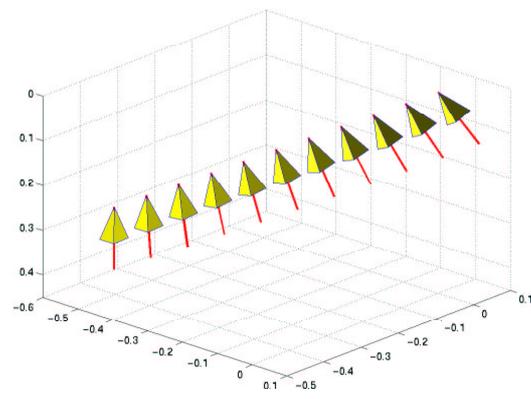


(b) trajectoire de la caméra

FIG. 5.19 – 20 % d'erreur sur les paramètres de calibration

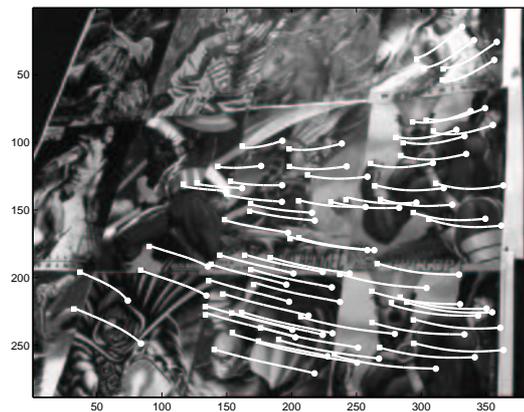


(a) trajectoires dans l'image

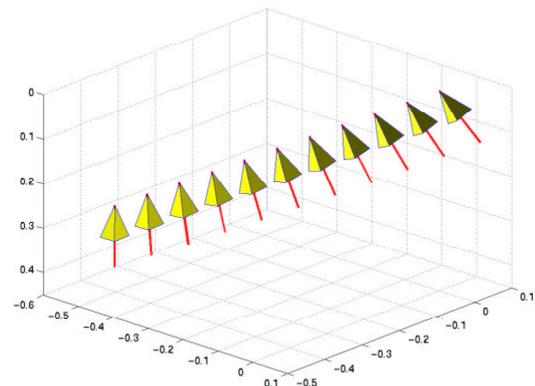


(b) trajectoire de la caméra

FIG. 5.20 – 40 % d'erreur sur les paramètres de calibration



(a) trajectoires dans l'image

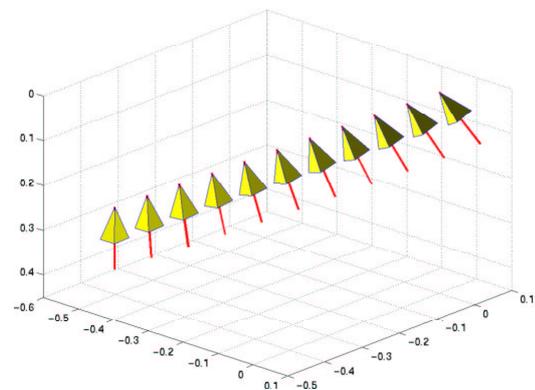


(b) trajectoire de la caméra

FIG. 5.21 – 60 % d'erreur sur les paramètres de calibration



(a) trajectoires dans l'image



(b) trajectoire de la caméra

FIG. 5.22 – 80 % d'erreur sur les paramètres de calibration

5.1.6 Application à l'asservissement visuel

Dans cette partie, nous nous plaçons dans le cadre général où les six degrés de liberté de la caméra sont à contrôler. Les primitives visuelles considérées pour contrôler le déplacement du robot sont les coordonnées de n points dans l'image, $\mathbf{s} = [x^1 \ y^1(t) \cdots x^n \ y^n(t)]$. Nous utiliserons les résultats de la section 5.1.3 pour planifier leurs trajectoires dans l'image. La loi de commande obtenue est composée de la somme d'une compensation proportionnelle de l'erreur et d'un terme de compensation de l'erreur de traînage.

5.1.6.1 Suivi des trajectoires par asservissement visuel 2D

Pour suivre la trajectoire en utilisant un asservissement visuel dans l'image, nous utilisons un asservissement visuel 2D. Nous définissons pour cela la fonction de tâche \mathbf{e} à réguler à $\mathbf{0}$ de la manière suivante :

$$\mathbf{e} = \widehat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}^*(t))$$

Dans la suite de cette partie, la notation $x^*(t)$ sera utilisée pour signifier la valeur de x à l'instant t obtenue à partir des primitives planifiées comme indiquées dans les sections précédentes. La notation x^f signifiera que l'on considère la valeur de x pour la position désirée de la caméra (i.e. à l'instant $t = 1$). Ainsi le vecteur $\mathbf{s}^*(t)$ est la trajectoire désirée des primitives visuelles dans l'image, et \mathbf{s}^f est le vecteur de primitives visuelles extraites de l'image finale. La matrice $\widehat{\mathbf{L}}^+$ est la pseudo-inverse d'un modèle choisi de la matrice d'interaction \mathbf{L} associée à \mathbf{s} . La matrice d'interaction dépend de la profondeur $Z^j(t)$ de chaque point dans le repère associé à la caméra à l'instant t (se référer à la relation (3.18)). Ce paramètre ne peut pas être obtenu directement. Cependant, le ratio $\rho^j(t) = Z^j(t)/d^f$ peut être estimé en utilisant la relation (2.35). Nous écrivons donc la matrice d'interaction associée à n points de la façon suivante :

$$\mathbf{L}(\mathbf{s}(t), \boldsymbol{\Sigma}(t), d^f) = \left[\frac{1}{d^f} \mathbf{S} \quad \mathbf{Q} \right] \quad (5.31)$$

où $\boldsymbol{\Sigma}(t) = [\rho^1(t) \cdots \rho^n(t)]$, et où $\mathbf{S} = [\mathbf{S}^{1T} \cdots \mathbf{S}^{nT}]^T$ et $\mathbf{Q} = [\mathbf{Q}^{1T} \cdots \mathbf{Q}^{nT}]^T$ sont deux matrices de dimensions $2n \times 3$ indépendantes du paramètre d^f :

$$\left\{ \begin{array}{l} \mathbf{S}^j = \mathbf{a} \begin{bmatrix} -\frac{1}{\rho^j(t)} & 0 & \frac{x^j(t)}{\rho^j(t)} \\ 0 & -\frac{1}{\rho^j(t)} & \frac{y^j(t)}{\rho^j(t)} \end{bmatrix} \\ \mathbf{Q}^j = \mathbf{a} \begin{bmatrix} x^j(t)y^j(t) & -1 - x^j(t)^2 & y^j(t) \\ 1 + y^j(t)^2 & -x^j(t)y^j(t) & -x^j(t) \end{bmatrix} \end{array} \right.$$

Nous choisissons alors :

$$\widehat{\mathbf{L}} = \mathbf{L}(\mathbf{s}^*(t), \widehat{\boldsymbol{\Sigma}}^*(t), \widehat{d}^f)$$

où \widehat{d}^f est le seul paramètre à fixer par l'utilisateur, et où $\widehat{\boldsymbol{\Sigma}}^*(t)$ peut être estimée comme nous le décrivons dans la suite. Comme nous l'avons déjà vu, une décroissance exponentielle de la fonction de tâche \mathbf{e} vers $\mathbf{0}$ peut être spécifiée en imposant $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ (λ représentant la constante de temps de la décroissance exponentielle), la loi de commande correspondante s'écrit alors :

$$\boldsymbol{\tau}_c = -\lambda\mathbf{e} - \frac{\partial\mathbf{e}}{\partial t}$$

où $\boldsymbol{\tau}_c$ est la vitesse de la caméra envoyée aux contrôleurs du robot. En outre, si la cible est immobile, alors nous avons $\frac{\partial\mathbf{e}}{\partial t} = -\widehat{\mathbf{L}}^+ \frac{\partial\mathbf{s}^*}{\partial t}$, et la commande en vitesse est donnée par la relation :

$$\boldsymbol{\tau}_c = -\lambda\mathbf{e} + \widehat{\mathbf{L}}^+ \frac{\partial\mathbf{s}^*}{\partial t}$$

où le terme $\widehat{\mathbf{L}}^+ \frac{\partial\mathbf{s}^*}{\partial t}$ permet de compenser les erreurs de traînage. Dans les paragraphes suivants, nous décrivons comment $\widehat{\boldsymbol{\Sigma}}^*(t)$ et $\frac{\partial\mathbf{s}^*}{\partial t}$ peuvent être obtenus.

Estimation de $\boldsymbol{\Sigma}^*(t)$. Nous rappelons que $\boldsymbol{\Sigma}^*(t) = [\rho^{*1}(t) \cdots \rho^{*n}(t)]$. Nous estimons $\rho^{*i}(t)$ en utilisant les relations (2.35) :

$$\left\{ \begin{array}{ll} \widehat{\rho}^{*i}(t) = \frac{\widehat{r}^*(t)}{\widehat{\mathbf{n}}^{*T}(t)\widehat{\mathbf{m}}^i(t)} & \text{si } \mathcal{M}^i \in \Pi \\ \widehat{\rho}^{*i}(t) = \widehat{\tau} \frac{\|\widehat{\mathbf{t}}_{df}\|}{\|\widehat{\mathbf{t}}/Z_f^i\|} & \text{si } \mathcal{M}^i \notin \Pi \end{array} \right. \quad (5.32)$$

Si la calibration n'est pas parfaite, la matrice $\widehat{\mathbf{K}}$ est utilisée à la place de \mathbf{K} . Alors la matrice d'homographie obtenue en présence d'erreurs de calibration est donnée en fonction de celle obtenue si la calibration est parfaite par :

$$\widehat{\mathbf{H}}^*(t) = \widehat{\mathbf{K}}^+ \mathbf{G}^*(t) \widehat{\mathbf{K}} = \alpha^*(t) \delta \mathbf{K} \mathbf{H}^*(t) \delta \mathbf{K}^+ \quad (5.33)$$

où $\alpha^*(t)$ est donné par (2.38) :

$$\alpha^*(t) = (\det(\widehat{\mathbf{H}}_\infty^*(t)))^{\frac{1}{3}} \quad (5.34)$$

En outre, d'après la relation (5.26), (5.34) peut s'écrire :

$$\alpha^*(t) = \det((\mathbf{G}_0 - \boldsymbol{\Phi}_0) \boldsymbol{\Gamma}^*(t))^{\frac{1}{3}} \quad (5.35)$$

or $\det(\boldsymbol{\Gamma}^*(t)) = 1$ implique :

$$\alpha^*(t) = \det((\mathbf{G}_0 - \boldsymbol{\Phi}_0))^{\frac{1}{3}} \quad (5.36)$$

D'après la relation précédente, $\alpha^*(t)$ est en fait indépendant du temps, nous le noterons donc dans la suite α^* . De plus comme \mathbf{G}_0 et Φ_0 peuvent être obtenus sans erreur, il en est de même pour α^* . Le paramètre $r^*(t)$ peut également être estimé sans erreur car :

$$r^*(t) = \det(\mathbf{H}^*(t)) = \det(\mathbf{G}^*(t)/\alpha) \quad (5.37)$$

Dans le cas considéré, nous avons la forme analytique de $\mathbf{G}^*(t)$, nous utiliserons donc la formulation suivante pour l'estimation de $r^*(t)$:

$$r^*(t) = \frac{\det(\mathbf{G}^*(t))}{\det(\mathbf{G}_0 - \Phi_0)} \quad (5.38)$$

D'autre part, la normale au plan de référence Π exprimée dans le repère associé à la caméra à l'instant t est donnée par :

$$\mathbf{n}^*(t) = \mathbf{R}^*(t)\mathbf{n}^f \quad (5.39)$$

et peut être estimée en présence d'erreurs de calibration de la manière suivante :

$$\hat{\mathbf{n}}^*(t) = \frac{1}{\alpha} \hat{\mathbf{H}}_{\infty}^{*-T}(t) \hat{\mathbf{n}}^f = \frac{1}{\alpha} \hat{\mathbf{K}}^{-1} (\mathbf{G}_0 - \Phi_0) \Gamma \hat{\mathbf{K}} \hat{\mathbf{n}}^f \quad (5.40)$$

En utilisant cette approximation pour le vecteur normal et dans le cas où le point considéré appartient au plan de référence, on aboutit à :

$$\hat{\mathbf{n}}^*(t) = \frac{1}{\alpha} \hat{\mathbf{H}}_{\infty}^{*-T}(t) \hat{\mathbf{n}}^f = (\delta \mathbf{K}^{-T} \mathbf{R}^* \delta \mathbf{K}^T) \left(\frac{\delta \mathbf{K}^{-T} \mathbf{n}^f}{\|\delta \mathbf{K}^{-T} \mathbf{n}^f\|} \right) = \frac{\delta \mathbf{K}^{-T} \mathbf{n}^*(t)}{\|\delta \mathbf{K}^{-T} \mathbf{n}^f\|} \quad (5.41)$$

et donc :

$$\hat{\rho}^{*i}(t) = \frac{\hat{r}^*(t)}{\hat{\mathbf{n}}^{*T}(t) \hat{\mathbf{m}}_i^*(t)} = \frac{r^*(t)}{\frac{\mathbf{n}^{*T}(t) \delta \mathbf{K}^{-1}}{\|\delta \mathbf{K}^{-T} \mathbf{n}^f\|} \delta \mathbf{K} \mathbf{m}_i^*(t)} = \|\delta \mathbf{K}^{-T} \mathbf{n}^f\| \rho^{*i}(t) \quad (5.42)$$

Nous utiliserons la relation (5.42) pour estimer les paramètres $\rho^{*i}(t)$ intervenant dans la loi de commande. L'erreur multiplicative commise en utilisant cette relation pour estimer $\rho^{*i}(t)$, lorsque le point considéré appartient au plan de référence, est alors $\|\mathbf{n}^{fT} \delta \mathbf{K}^{-1}\|$. En remarquant, de plus, que la translation partielle peut être estimée de la manière suivante :

$$\hat{\mathbf{t}}_{df}(t) = (1 - q(t)) \hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f \quad (5.43)$$

on aboutit dans le cas où le point considéré n'appartient pas au plan de référence à :

$$\hat{\rho}^{*i}(t) = (1 - q(t)) \hat{\tau}(t) \frac{\|\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f\|}{\|\hat{\tau}(t) \hat{\mathbf{K}}^+ \mathbf{p}_i^f - \hat{\mathbf{K}}^+ (\mathbf{G}_0 - \Phi_0) \Gamma^*(t) \mathbf{p}_i^*(t)\|} \quad (5.44)$$

avec :

$$\hat{\tau}(t) = \frac{\|[\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f]_{\times} (\mathbf{G}_0 - \Phi_0) \Gamma^*(t) \mathbf{p}_i^f\|}{\|[\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f]_{\times} \hat{\mathbf{K}}^+ \mathbf{p}_i^*(t)\|} \quad (5.45)$$

Estimation de $\frac{\partial \mathbf{s}^*}{\partial t}$. La relation (5.27) conduit à l'expression suivante pour la dérivée temporelle de $\mathbf{p}^*(t)$:

$$\dot{\mathbf{p}}_i^*(t) = \frac{1}{\alpha_i^*(t)} \left[\dot{\mathbf{G}}^*(t) \mathbf{p}_i^f + \beta_i \dot{\mathbf{e}}^*(t) - \dot{\alpha}_i^*(t) \mathbf{p}_i^*(t) \right] \quad (5.46)$$

et si nous écrivons la matrice de colinéation et l'épipole sous la forme :

$$\mathbf{G}(t) = \begin{bmatrix} \mathbf{G}_1(t) \\ \mathbf{G}_2(t) \\ \mathbf{G}_3(t) \end{bmatrix}, \quad \mathbf{e}(t) = \begin{bmatrix} \mathbf{e}_1(t) \\ \mathbf{e}_2(t) \\ \mathbf{e}_3(t) \end{bmatrix} \quad (5.47)$$

nous avons :

$$\left\{ \begin{array}{l} \dot{\mathbf{G}}^*(t) = \dot{q}(t) [-\Phi_0 + (\mathbf{G}_0 - \Phi_0) \Gamma^*(t)] = \begin{bmatrix} \dot{\mathbf{G}}_1^*(t) \\ \dot{\mathbf{G}}_2^*(t) \\ \dot{\mathbf{G}}_3^*(t) \end{bmatrix} \\ \dot{\mathbf{e}}^*(t) = -\dot{q}(t) \mathbf{e}_0 = \begin{bmatrix} \dot{\mathbf{e}}_1^*(t) \\ \dot{\mathbf{e}}_2^*(t) \\ \dot{\mathbf{e}}_3^*(t) \end{bmatrix} \\ \alpha_i^*(t) = \mathbf{G}_3^*(t) \mathbf{p}_i^f + \beta_i \mathbf{e}_3^*(t) \\ \dot{\alpha}_i^*(t) = \dot{\mathbf{G}}_3^*(t) \mathbf{p}_i^f + \beta_i \dot{\mathbf{e}}_3^*(t) \end{array} \right. \quad (5.48)$$

En introduisant les relations données par (5.48) dans la relation (5.46), nous obtenons la dérivée temporelle de la trajectoire planifiée dans l'image $\dot{\mathbf{p}}_i^*(t) = [\dot{u}_i^*(t) \ \dot{v}_i^*(t) \ 0]$ de la projection de chaque point \mathcal{M}_i considéré. Nous en déduisons directement la valeur du vecteur de paramètres $\frac{\partial \mathbf{s}^*}{\partial t} = \dot{\mathbf{s}}^*(t) = [\dot{u}_1^*(t) \ \dots \ \dot{v}_n^*(t)]$.

5.1.7 Résultats expérimentaux

Les méthodes proposées ont été testées sur la plate-forme robotique de l'IRISA, équipée d'un robot cartésien AFMA à six degrés de liberté. La caméra est embarquée sur l'organe terminal du robot et observe un objet d'intérêt statique. Le modèle CAO de cet objet est supposé inconnu. La tâche consiste à repositionner le robot de telle sorte que l'image acquise par la caméra à la fin de l'asservissement corresponde à une image acquise lors d'une phase d'apprentissage de l'image désirée. Nous avons utilisé trois types de cible :

- un objet plan composé de quatre disques blancs sur fond noir (paragraphe 5.1.7.1);
- un objet non plan composé de neuf disques blancs sur fond noir (paragraphe 5.1.7.2);
- l'image d'une scène complexe (paragraphe 5.1.7.3).

Les erreurs de traînage, subsistant à la fin de la phase de suivi de trajectoires, sont éliminées en utilisant une commande proportionnelle à l'erreur entre le vecteur de mesure courante et le vecteur de consigne finale désirée.

5.1.7.1 Cible plane

Dans ce paragraphe, nous présentons les résultats obtenus avec une cible plane. Les informations visuelles utilisées sont les coordonnées du centre de gravité de chaque disque. Les images initiale et désirée de la scène sont illustrées sur les figures 5.23(a) et 5.23(b) respectivement. Le déplacement entre ces deux prises de vue est très important et correspond à :

translation (mm)	$t_x = 28$	$t_y = 440$	$t_z = 1008$
rotation (dg)	$(u\theta)_x = -73$	$(u\theta)_y = -50$	$(u\theta)_z = 134$

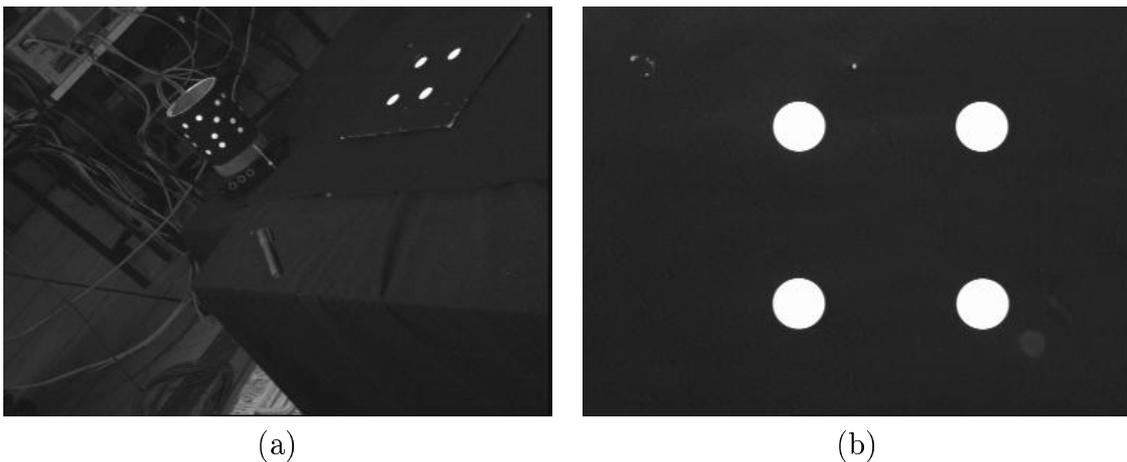


FIG. 5.23 – (a) Image initiale et (b) image désirée

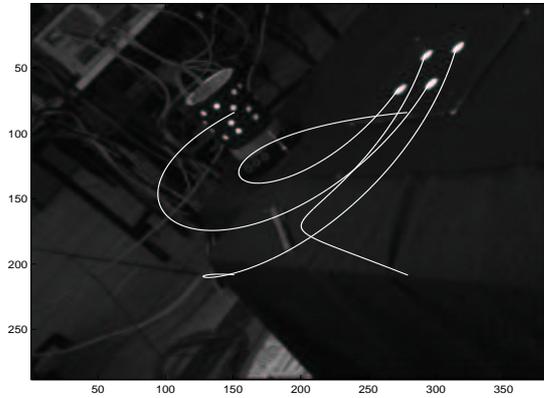
Pour vérifier la robustesse de notre méthode de planification de trajectoires, nous présentons les résultats obtenus en utilisant trois ensembles de paramètres :

1. calibration correcte : les paramètres intrinsèques sont correctement estimés et la profondeur d^f est prise égale à sa valeur réelle de 36 cm,
2. mauvaise calibration : dans ce cas, une erreur de 20% a été ajoutée à chacun des paramètres intrinsèques et la profondeur d^f a été prise égale à 20 cm,
3. très mauvaise calibration : dans ce dernier cas, une erreur de 50% a été ajoutée à chacun des paramètres intrinsèques et la profondeur d^f a été prise égale à 80 cm.

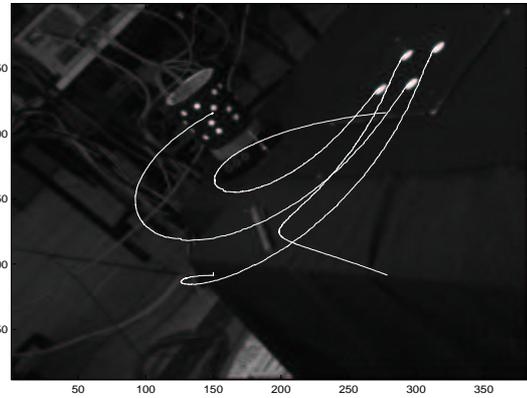
Pour chaque ensemble de paramètres, nous avons réalisé deux expériences : une expérience en utilisant la solution à minimum d'énergie et une seconde en utilisant la solution à minimum d'accélération.

1) **Calibration correcte** : Soulignons, tout d'abord, que les trajectoires dans l'image obtenues avec les solutions à minimum d'énergie et à minimum d'accélération sont identiques (comparer les figures 5.24(a) avec 5.25(a) et 5.25(b) avec 5.25(b)). Les trajectoires suivies sont fournies sur les figure 5.24(b) et 5.25(b). Il faut souligner que ces trajectoires sont similaires aux trajectoires planifiées illustrées sur les figures 5.24(a) et 5.25(a). Cela montre l'efficacité du schéma de commande. Les erreurs de suivi ($\mathbf{s}(t) - \mathbf{s}^*(t)$) sont représentées par les figures 5.24(d) et 5.25(d). On remarque que ces erreurs restent suffisamment faibles dans les deux cas (l'erreur maximale reste inférieure à 5 pixels) pour assurer le bon comportement de la loi de commande 2D. Les erreurs sur les coordonnées des points dans l'image sont représentées sur les figures 5.24(c) et 5.25(c). La convergence des coordonnées dans l'image de chacun des points considérés vers leur valeur désirée démontre que la tâche a été correctement réalisée. Les trajectoires de la caméra dans l'espace de travail sont illustrées sur les figures 5.24(f) et 5.25(f). Les chemins suivis par la caméra, dans les deux cas, sont identiques et très satisfaisants. Seul l'aspect temporel diffère. Cela se retrouve au niveau des courbes représentant la vitesse de la caméra données par les figures 5.24(e) et 5.25(e). La vitesse maximale est atteinte plus rapidement dans le cas du minimum d'énergie.

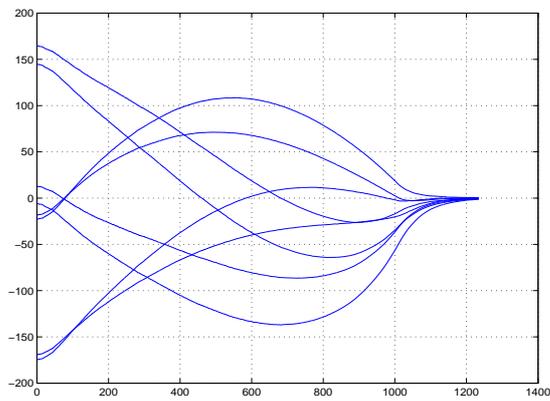
2) et 3) **Mauvaise et très mauvaise calibrations** : Nous testons ici la robustesse de notre méthode vis-à-vis des erreurs sur l'estimation de la calibration et de la profondeur d^f . Comme nous pouvons le constater sur les figures 5.24(a), 5.26(a) et 5.25(a), 5.27(a), les trajectoires planifiées dans les cas de la calibration correcte et de la mauvaise calibration sont similaires. Cela confirme, de nouveau, la robustesse de l'algorithme de planification de trajectoires vis-à-vis des erreurs considérées. Notons également que les trajectoires planifiées, dans le cas de la très mauvaise calibration, représentées sur les figures 5.28(a) et 5.29(a) sont également proches de celles obtenues dans les autres cas. Les trajectoires réalisées (voir figures 5.26(b), 5.27(b) et 5.28(b), 5.29(b)), sont elles aussi très satisfaisantes. Les erreurs de suivi fournies par les figures 5.26(d) et 5.27(d), dans le cas de la mauvaise calibration, restent faibles durant l'asservissement (toujours inférieures à 5 pixels). Dans le cas d'une très mauvaise calibration, les erreurs de suivi sont représentées par les figures 5.28(d) et 5.28(d) et restent toujours inférieures à 8 pixels. Dans ces configurations, l'asservissement visuel 2D fournit de très bons résultats et fait preuve d'une grande robustesse. Dans les deux expérimentations, la tâche de positionnement est correctement réalisée comme le démontre la convergence vers 0 de l'erreur des coordonnées dans l'image de tous les points considérés (voir figures 5.26(e), 5.27(e) et 5.28(e), 5.29(e)). Une nouvelle fois, les chemins suivis par la caméra, dans tous les cas, sont identiques et très satisfaisants. Seul l'aspect temporel diffère entre les solutions à minimum d'énergie et minimum d'accélération. Cela se retrouve également au niveau des courbes représentant la vitesse de la caméra.



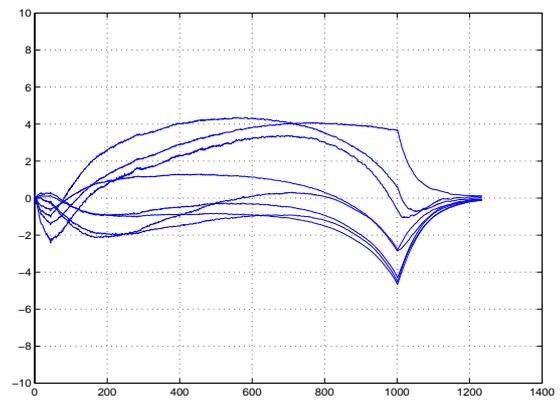
(a) trajectoires planifiées



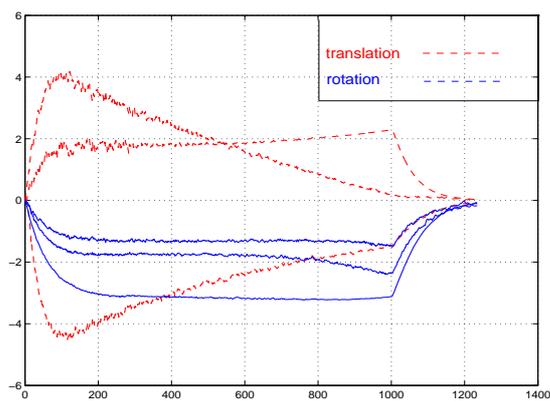
(b) trajectoires suivies



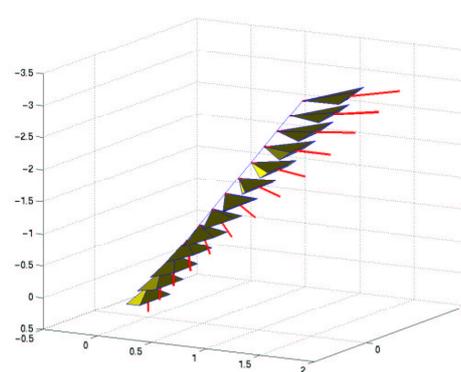
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

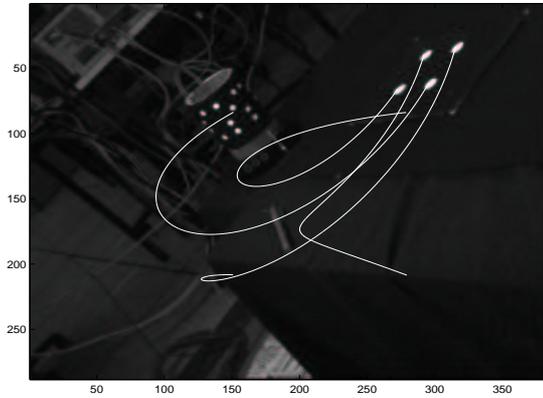


(e) vitesse de la caméra (cm/s et deg/s)

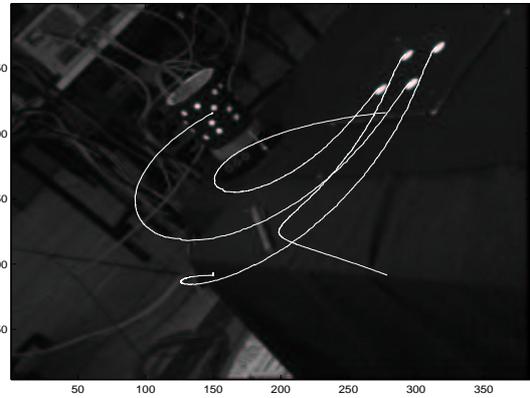


(f) trajectoire de la caméra

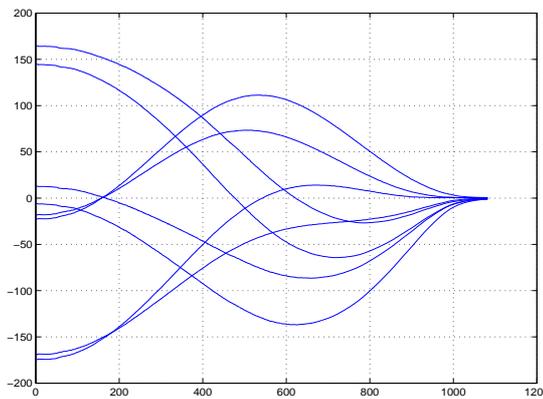
FIG. 5.24 – Calibration correcte : minimum d'énergie



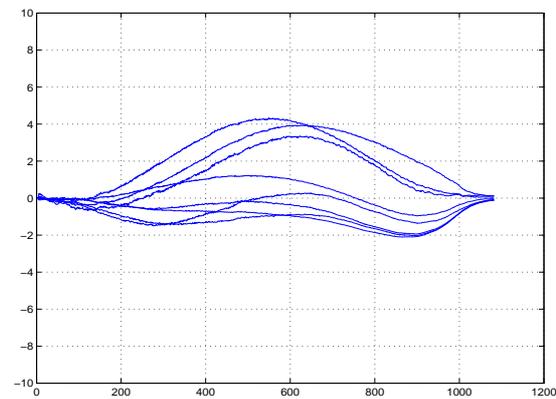
(a) trajectoires planifiées



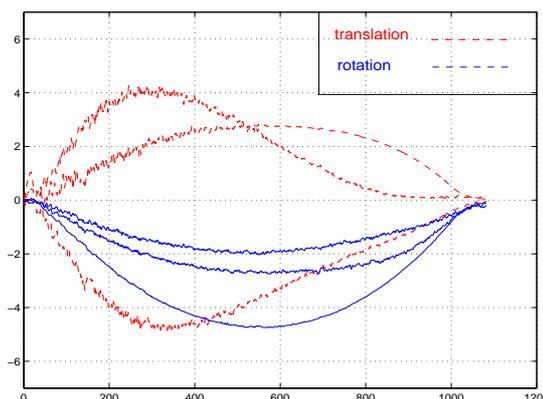
(b) trajectoires suivies



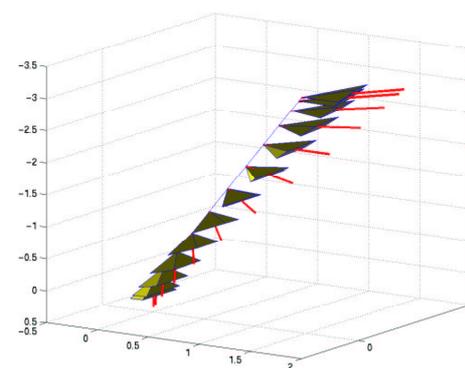
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

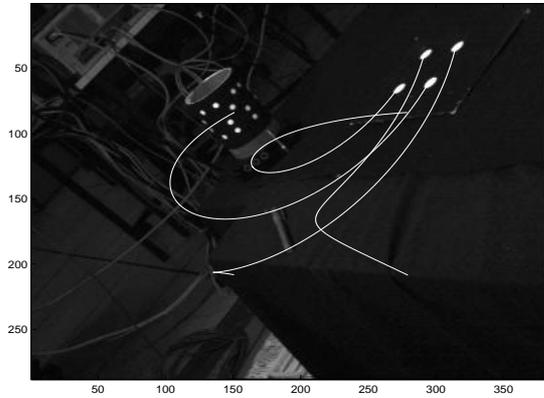


(e) vitesse de la caméra (cm/s et deg/s)

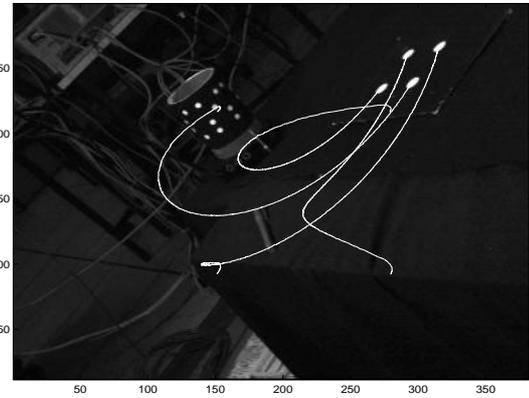


(f) trajectoire de la caméra

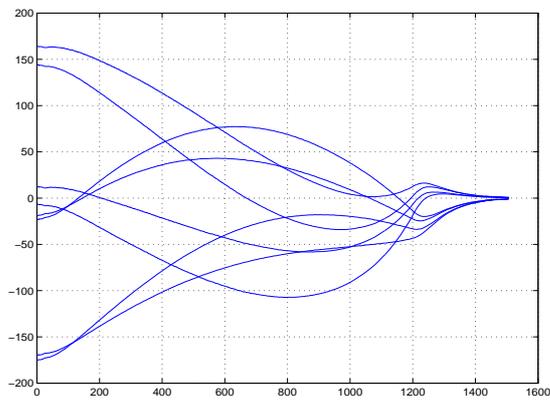
FIG. 5.25 – Calibration correcte : minimum d'accélération



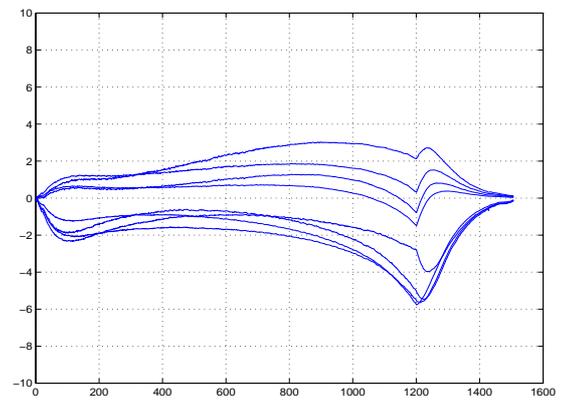
(a) trajectoires planifiées



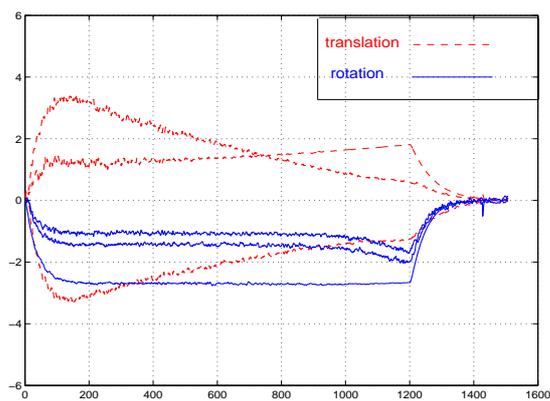
(b) trajectoires suivies



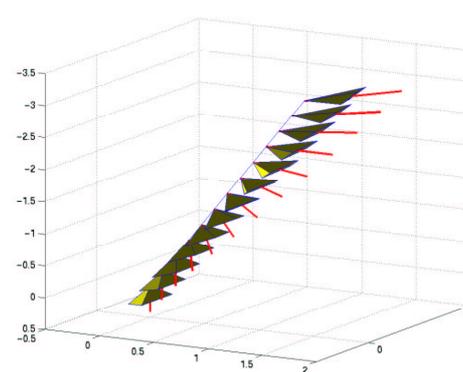
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

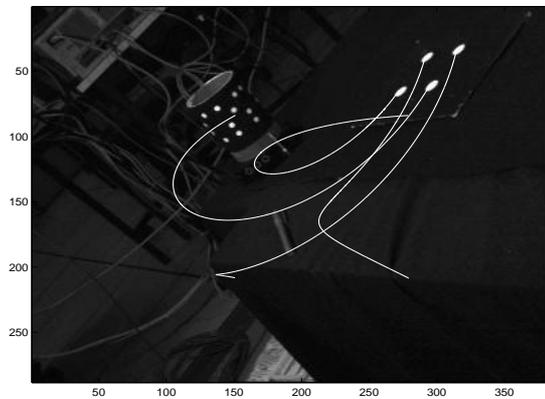


(e) vitesse de la caméra (cm/s et deg/s)

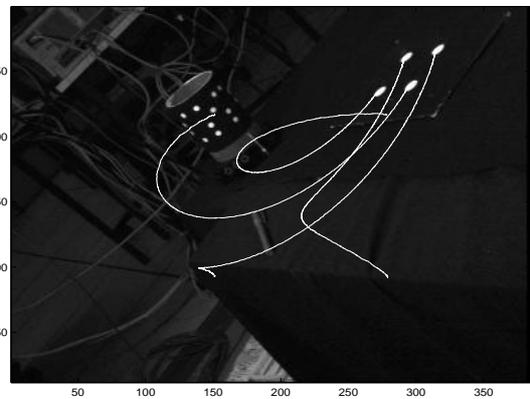


(f) trajectoire de la caméra

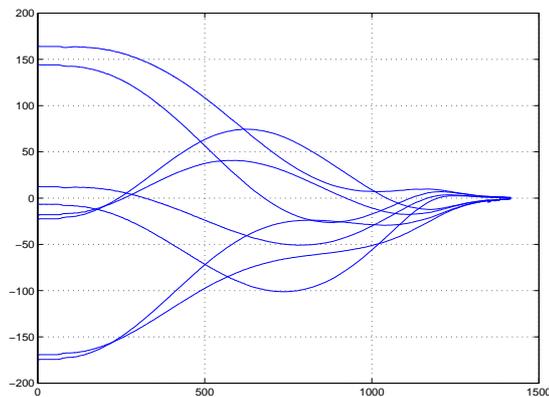
FIG. 5.26 – Mauvaise calibration : minimum d'énergie



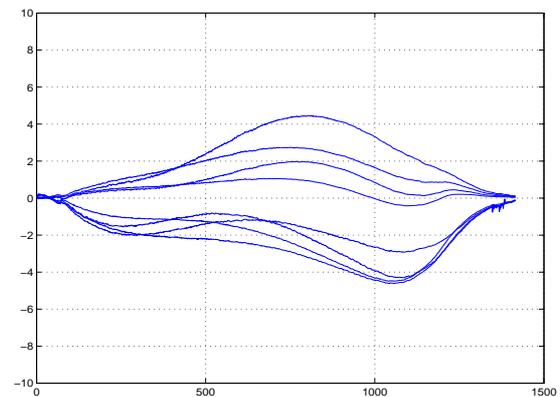
(a) trajectoires planifiées



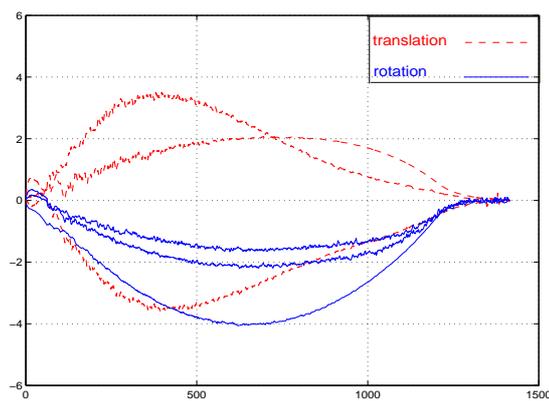
(b) trajectoires suivies



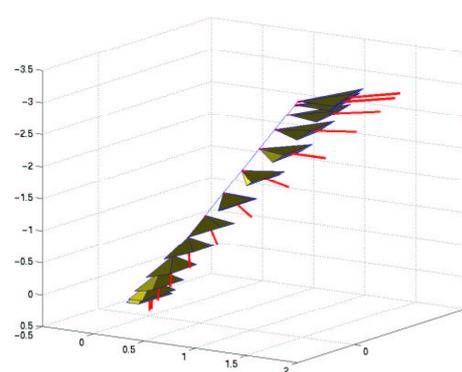
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

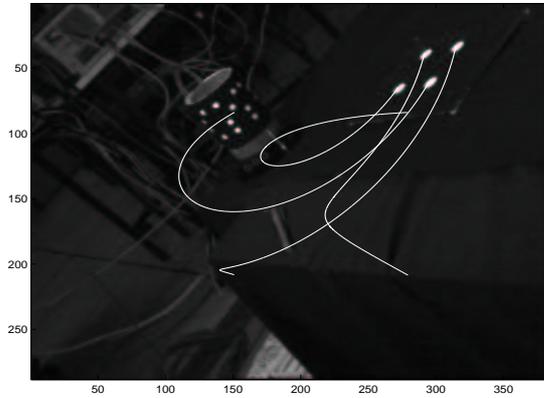


(e) vitesse de la caméra (cm/s et deg/s)

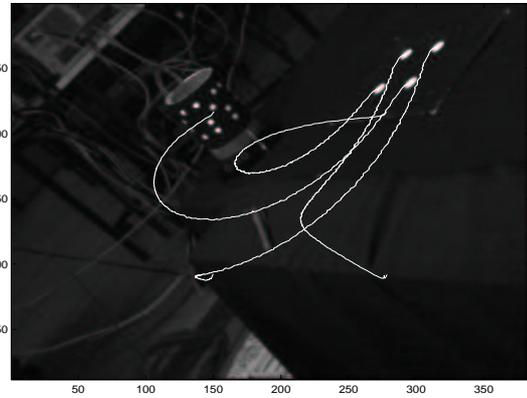


(f) trajectoire de la caméra

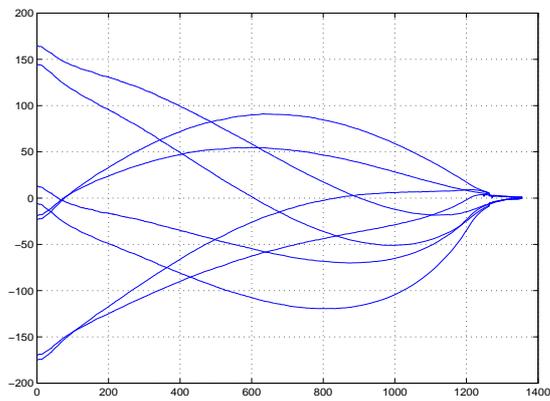
FIG. 5.27 – Mauvaise calibration : minimum d'accélération



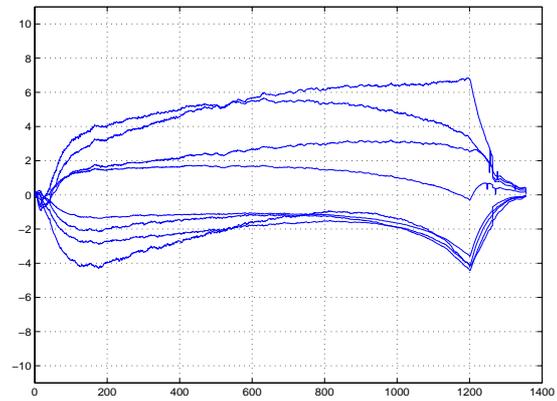
(a) trajectoires planifiées



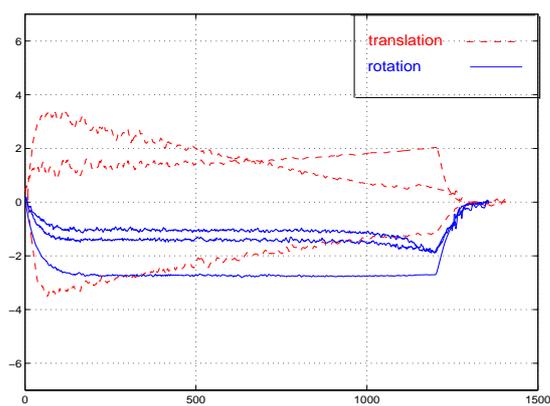
(b) trajectoires suivies



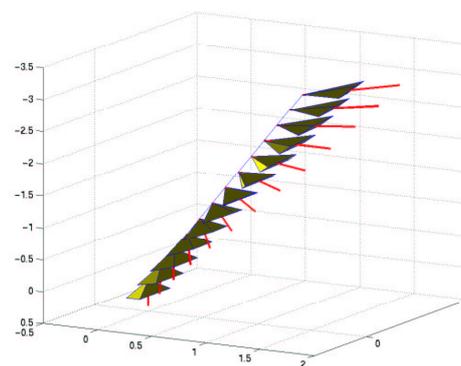
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

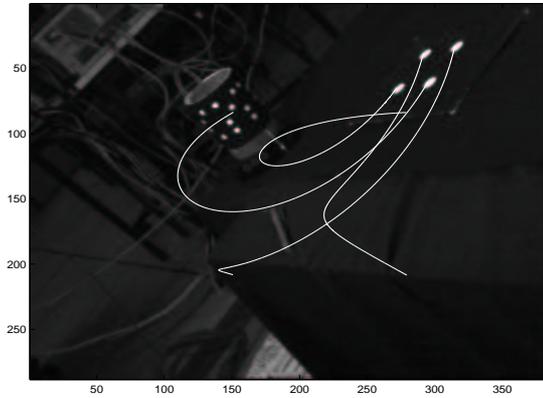


(e) vitesse de la caméra (cm/s et deg/s)

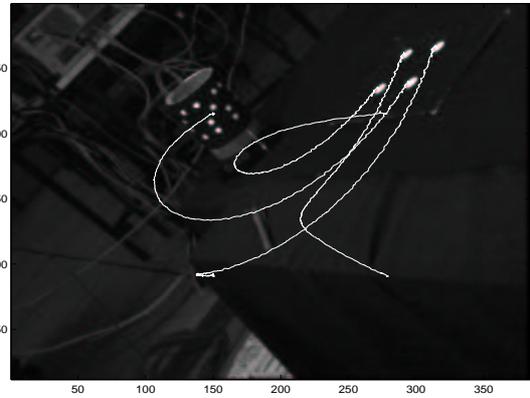


(f) trajectoire de la caméra

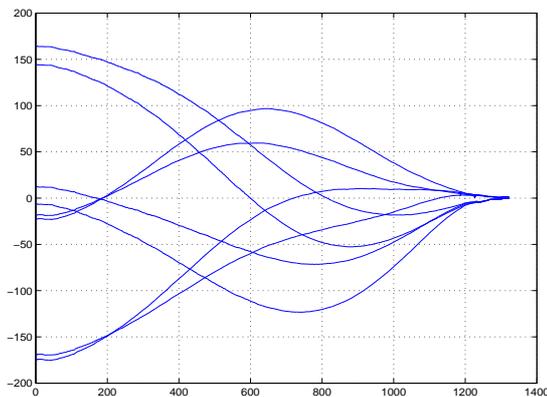
FIG. 5.28 – Très mauvaise calibration : minimum d'énergie



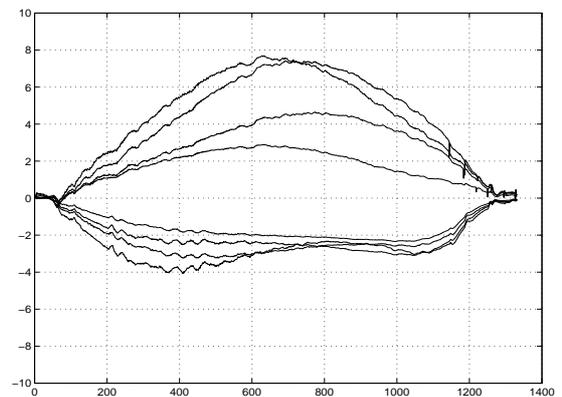
(a) trajectoires planifiées



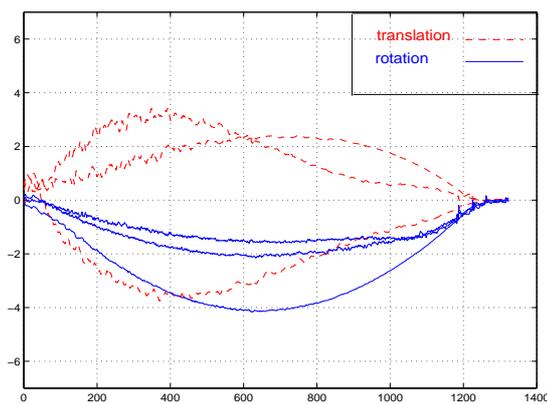
(b) trajectoires suivies



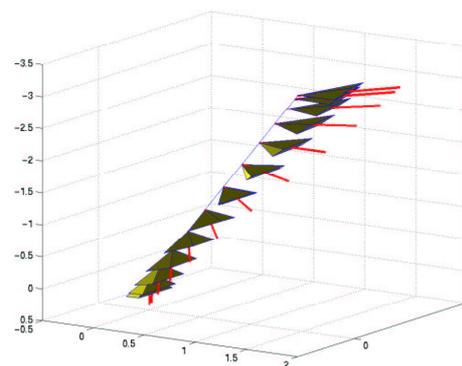
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel



(e) vitesse de la caméra (cm/s et deg/s)



(f) trajectoire de la caméra

FIG. 5.29 – Très mauvaise calibration : minimum d'accélération

5.1.7.2 Cible non-plane

La cible est maintenant non plane. De nouveau, le déplacement à effectuer est très important :

translation (mm)	$t_x = -195$	$t_y = -610$	$t_z = -1455$
rotation (dg)	$(u\theta)_x = -68$	$(u\theta)_y = -41$	$(u\theta)_z = -144$

Les trois ensembles de paramètres désignés par calibration correcte, mauvaise calibration et très mauvaise calibration ont également été utilisés. La profondeur d^f a été prise respectivement égale à 50cm, 20cm et 100cm dans les cas de calibration correcte, mauvaise calibration et très mauvaise calibration (alors que la valeur réelle est de l'ordre de 50cm). Dans cette partie, nous nous contentons de présenter les résultats car leur description est identique à celle des cas où la cible était plane. Notons cependant que les trajectoires obtenues, à la fois de la caméra dans l'espace cartésien et de l'objet dans l'image, sont particulièrement satisfaisantes dans les trois cas.

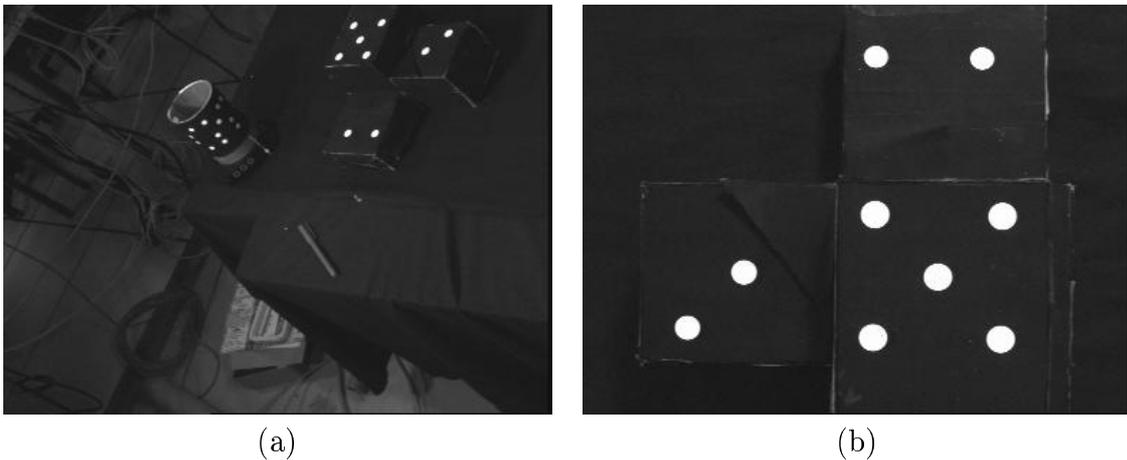
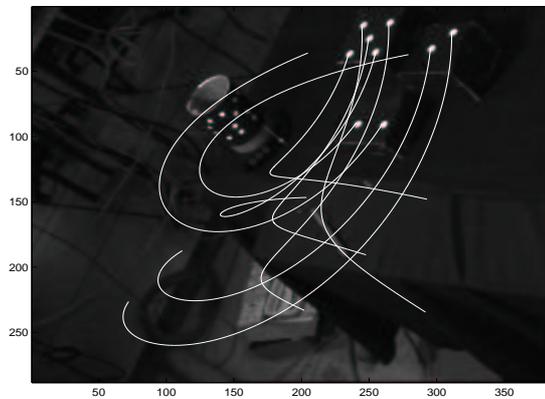
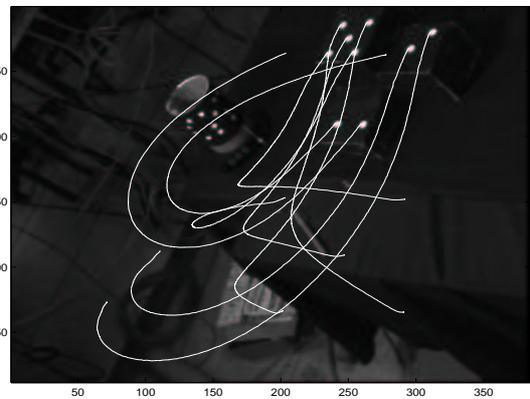


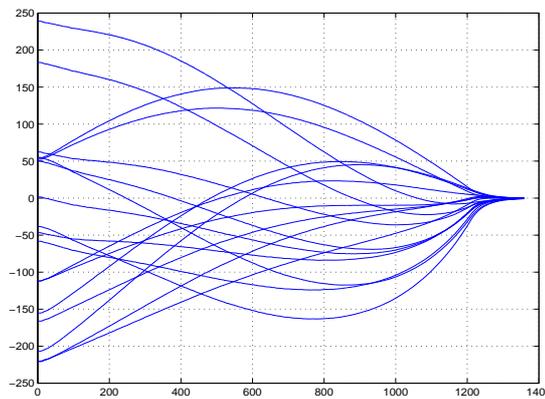
FIG. 5.30 – (a) Image initiale et (b) image désirée



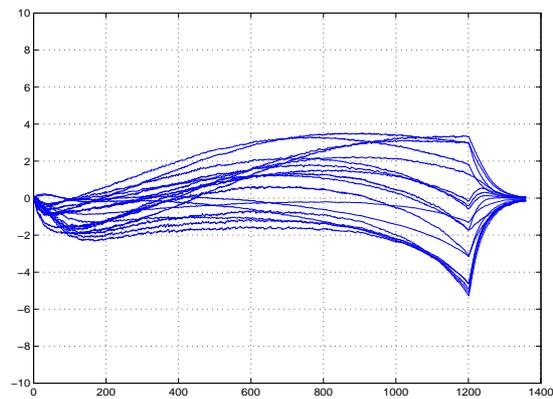
(a) trajectoires planifiées



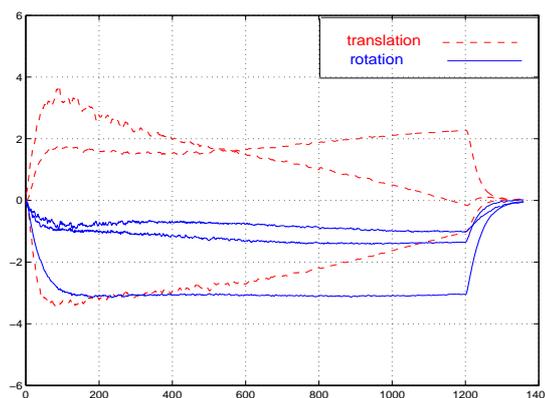
(b) trajectoires suivies



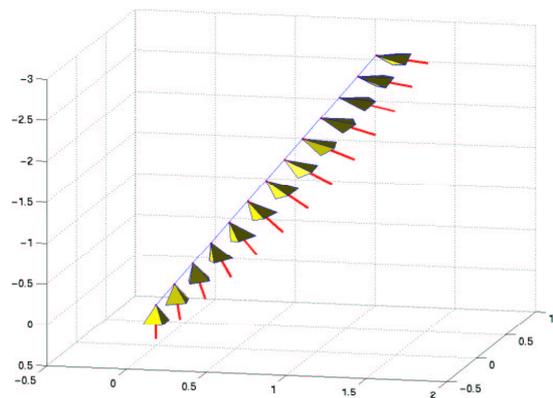
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel



(e) vitesse de la caméra (cm/s et deg/s)



(f) trajectoire de la caméra

FIG. 5.31 – Calibration correcte : minimum d'énergie

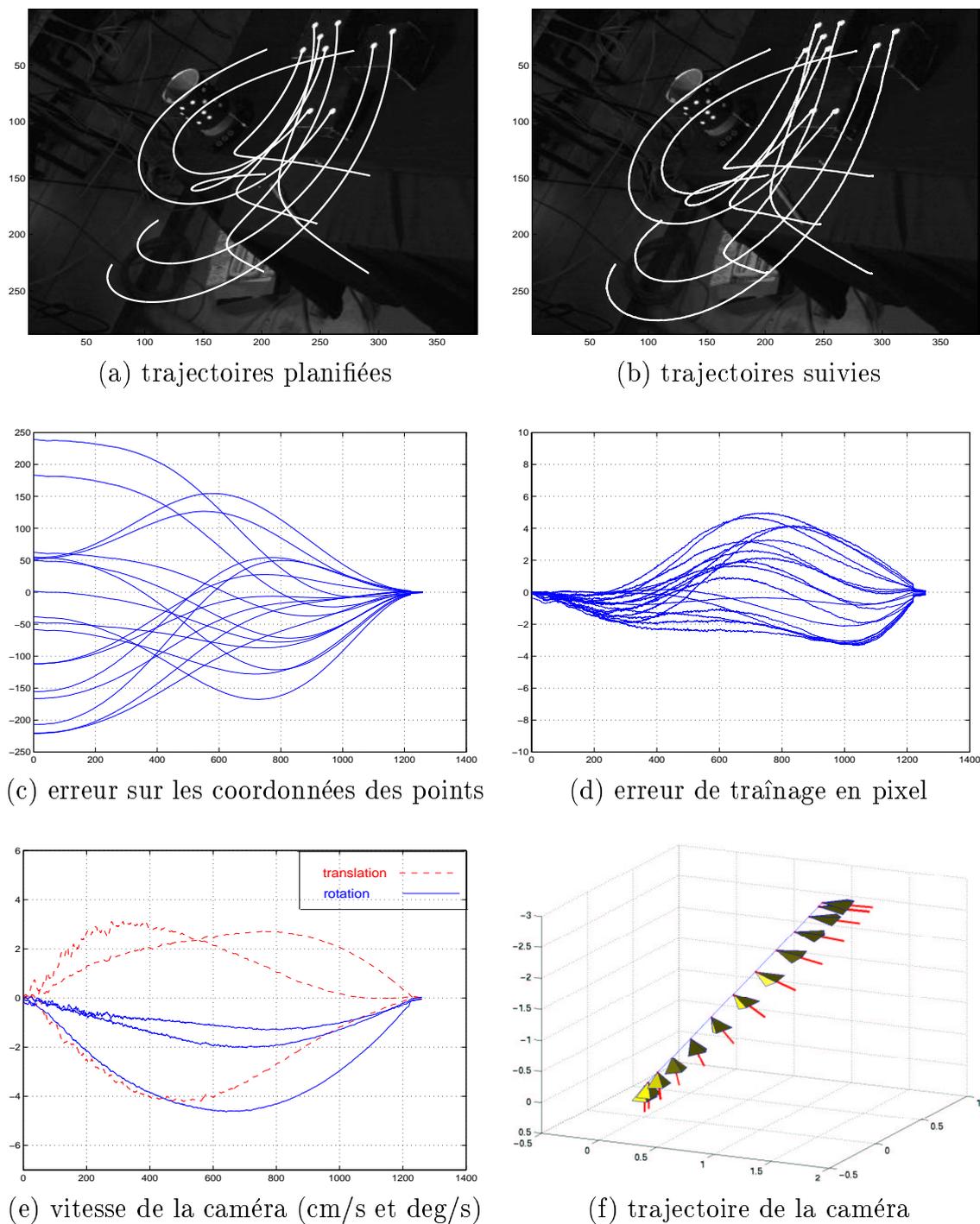
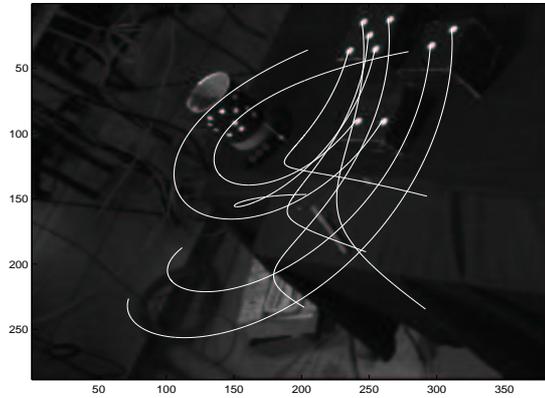
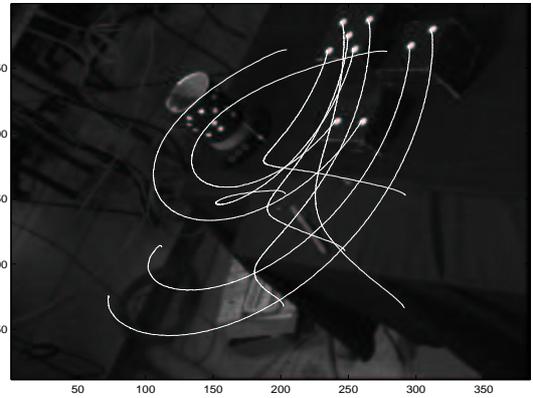


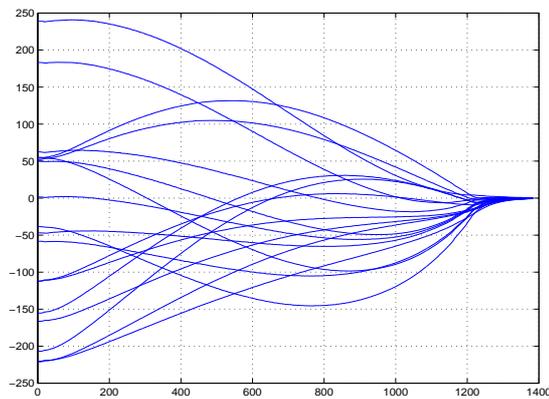
FIG. 5.32 – Calibration correcte : minimum d'accélération



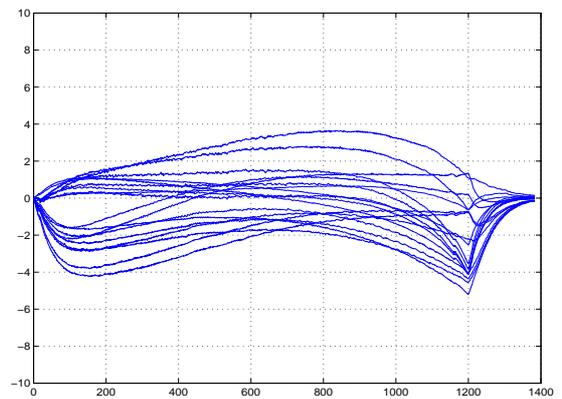
(a) trajectoires planifiées



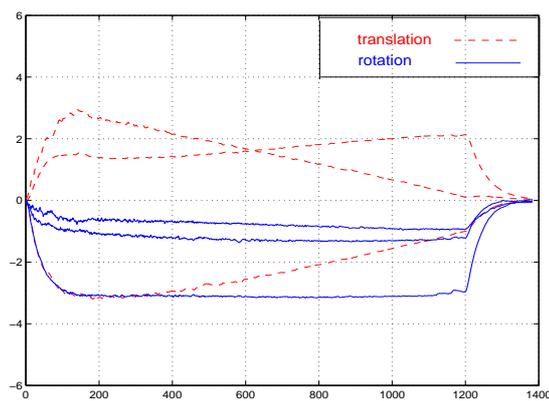
(b) trajectoires suivies



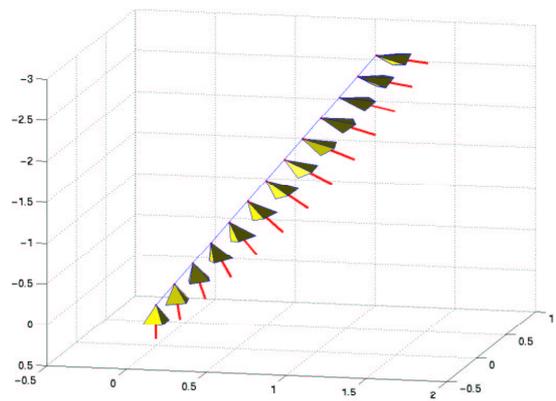
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

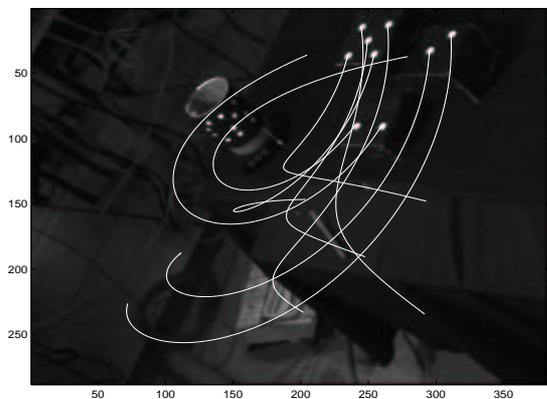


(e) vitesse de la caméra (cm/s et deg/s)

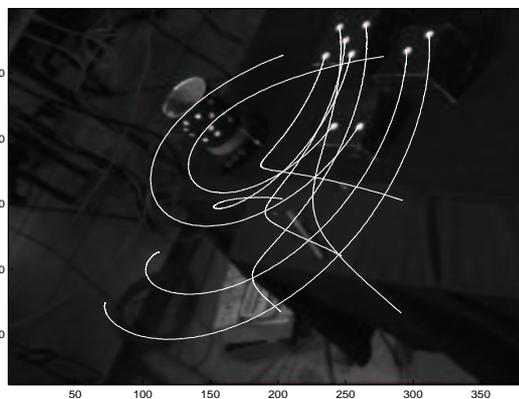


(f) trajectoire de la caméra

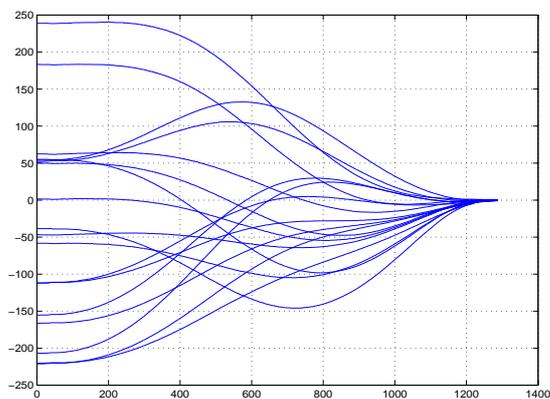
FIG. 5.33 – Mauvaise calibration : minimum d'énergie



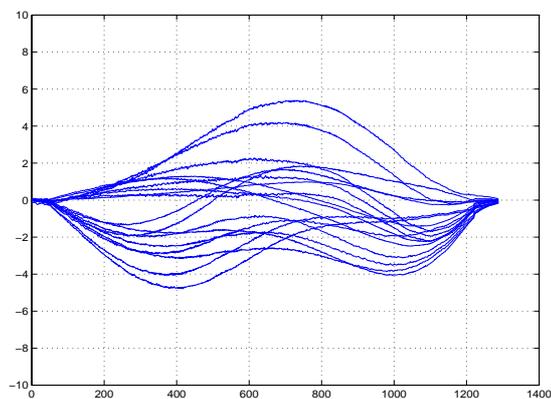
(a) trajectoires planifiées



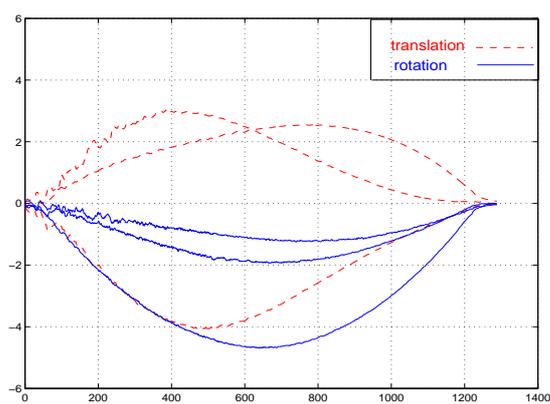
(b) trajectoires suivies



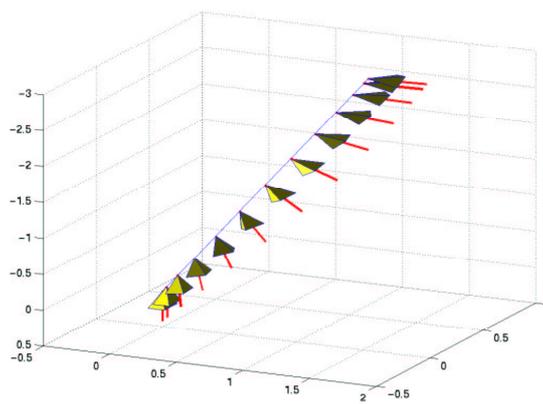
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

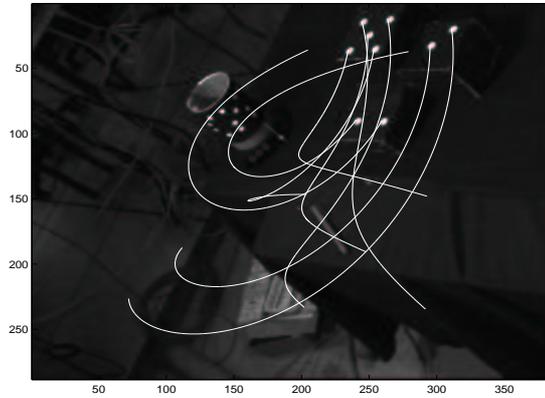


(e) vitesse de la caméra (cm/s et deg/s)

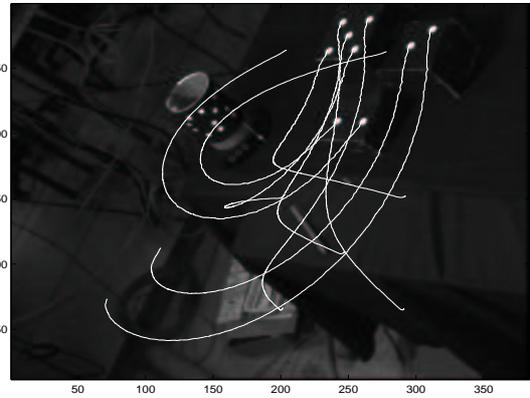


(f) trajectoire de la caméra

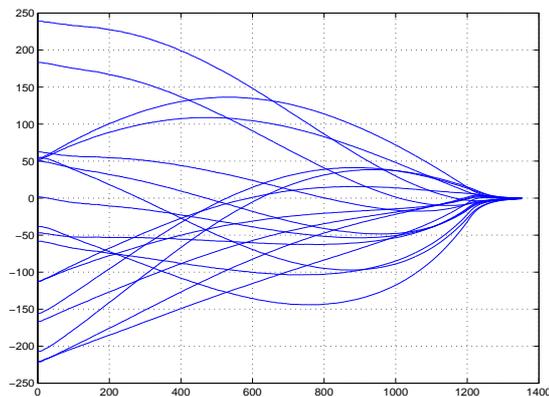
FIG. 5.34 – Mauvaise calibration : minimum d'accélération



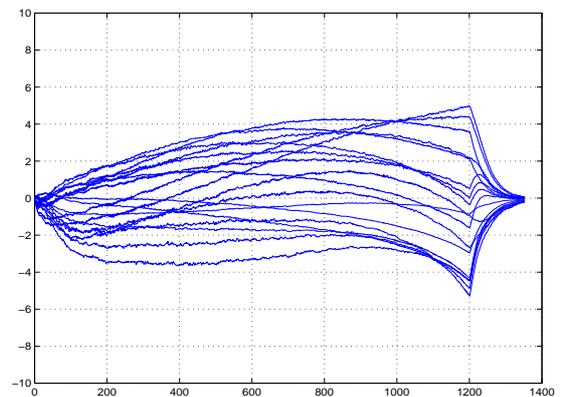
(a) trajectoires planifiées



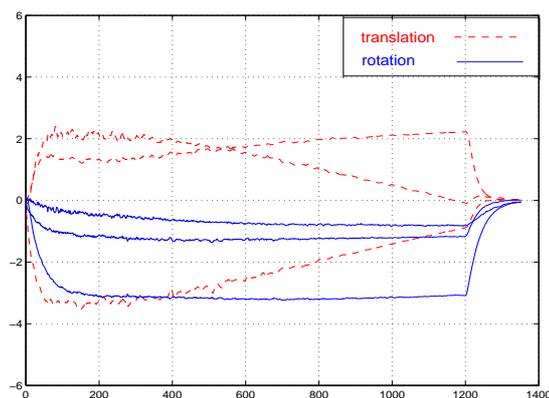
(b) trajectoires suivies



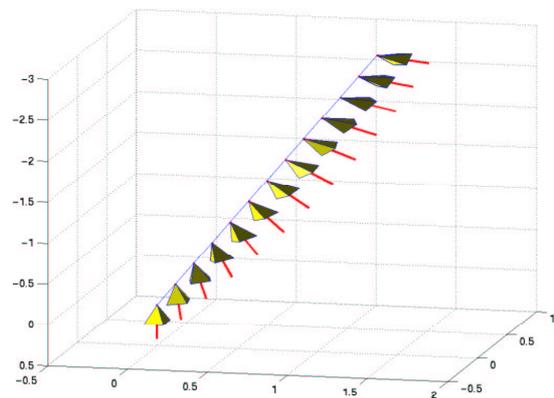
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel



(e) vitesse de la caméra (cm/s et deg/s)



(f) trajectoire de la caméra

FIG. 5.35 – Très mauvaise calibration : minimum d'énergie

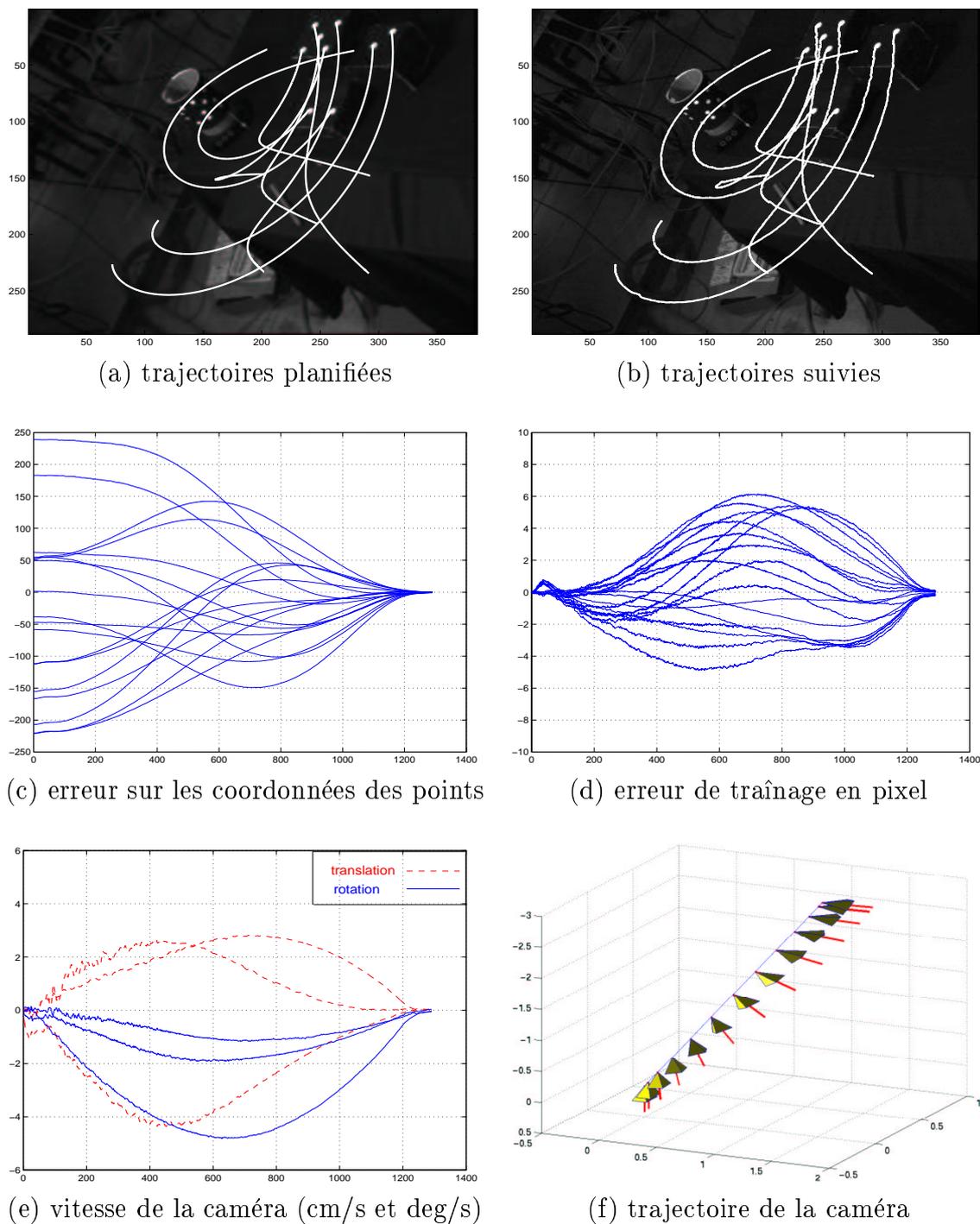


FIG. 5.36 – *Très mauvaise calibration : minimum d'accélération*

5.1.7.3 Scène complexe

Dans les deux expériences que nous présentons à présent, les primitives visuelles extraites sont les points d'intérêt obtenus en utilisant le détecteur de Harris [Harris 88]. Les points d'intérêt sont mis en correspondance en utilisant l'algorithme présenté dans [Zhang 95a]. Ces points sont représentés par des croix dans les images initiale et désirée illustrées sur les figures 5.37(a) et 5.37(b). Pour vérifier la robustesse vis-à-vis des erreurs sur les paramètres de calibration et sur d^f , nous avons réalisé deux expériences avec des ensembles de paramètres différents :

1. dans l'expérience présentée par la figure 5.38, nous avons utilisé les paramètres de calibration fournis par le constructeur de la caméra et une approximation correcte de la profondeur d^f (i.e $d^f = 0.9\text{m}$),
2. dans la seconde expérience présentée par la figure 5.39, une erreur de 30% a été ajoutée aux paramètres intrinsèques et la profondeur d^f a été prise égale à 0.5m.

Comme nous pouvons le constater sur les figures 5.38(a), 5.38(b) et 5.39(a), 5.39(b), les trajectoires planifiées et suivies sont similaires dans les deux cas. Les trajectoires obtenues avec des paramètres corrects sont proches de celles obtenues avec les mauvais paramètres (comparer la figure 5.38(a) avec la figure 5.39(a) et la figure 5.38(b) avec la figure 5.39(b)). Cela confirme, de nouveau, la robustesse de la phase de planification et d'asservissement vis-à-vis des erreurs sur la calibration de la caméra et sur la profondeur d^f . Les erreurs de suivi définies par $\frac{1}{2n} \sum_1^n (x_i(t) - x_i^*(t)) + (y_i(t) - y_i^*(t))$ et illustrées sur les figures 5.38(d) et 5.39(d) restent faibles dans les deux cas (toujours inférieures à 5 pixels). Finalement, la convergence vers 0 des erreurs moyennes, entre les coordonnées courantes et finales des points d'intérêt définies par $\frac{1}{2n} \sum_1^n (x_i(t) - x_i^*(1)) + (y_i(t) - y_i^*(1))$ et représentées sur les figures 5.38(c) et 5.39(c), démontre que la tâche de positionnement a été correctement réalisée dans les deux cas de figure.

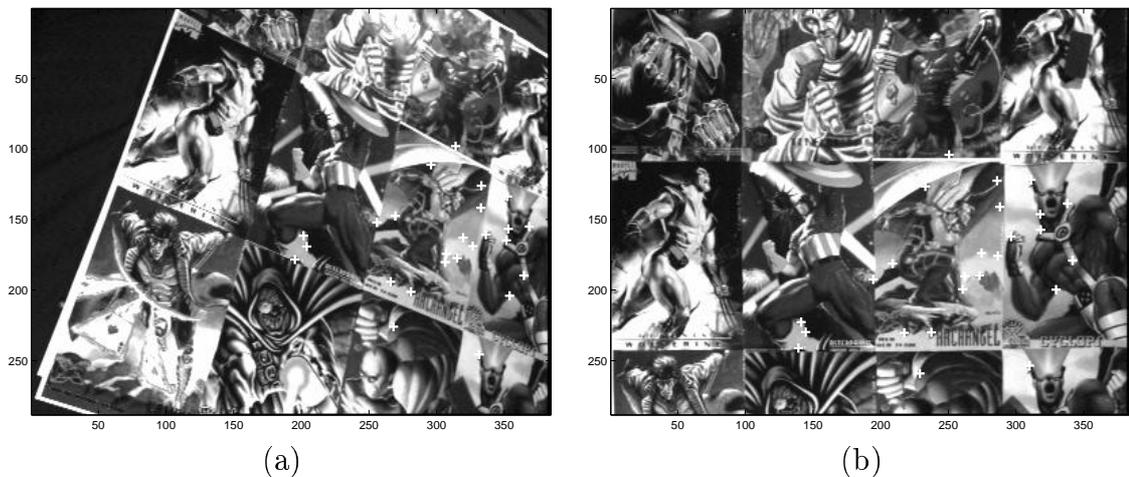


FIG. 5.37 – (a) Image initiale et (b) image désirée

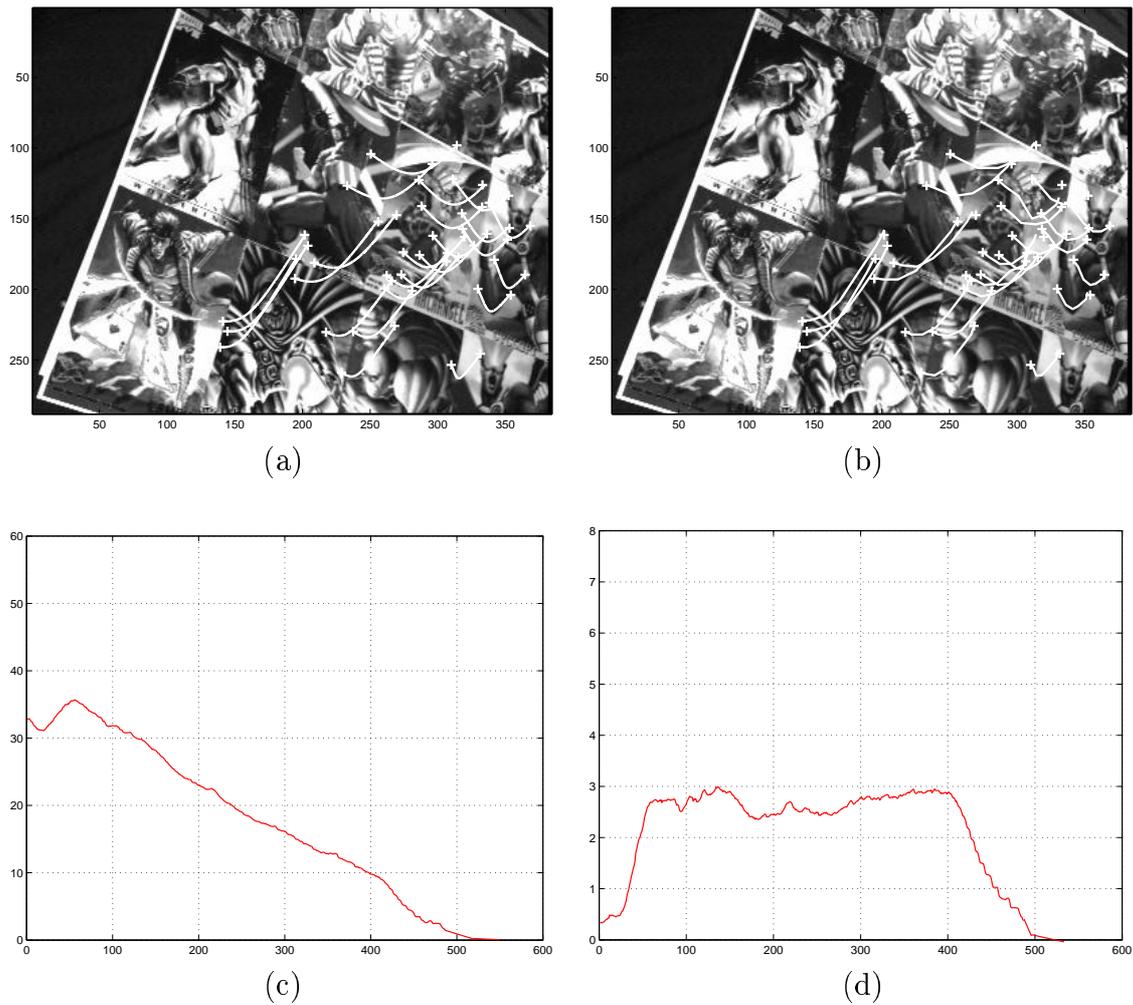


FIG. 5.38 – Calibration correcte (a) trajectoires planifiées, (b) trajectoires suivies, (c) erreurs par rapport aux coordonnées désirées dans l'image (pixels), (d) erreur de suivi (pixels)

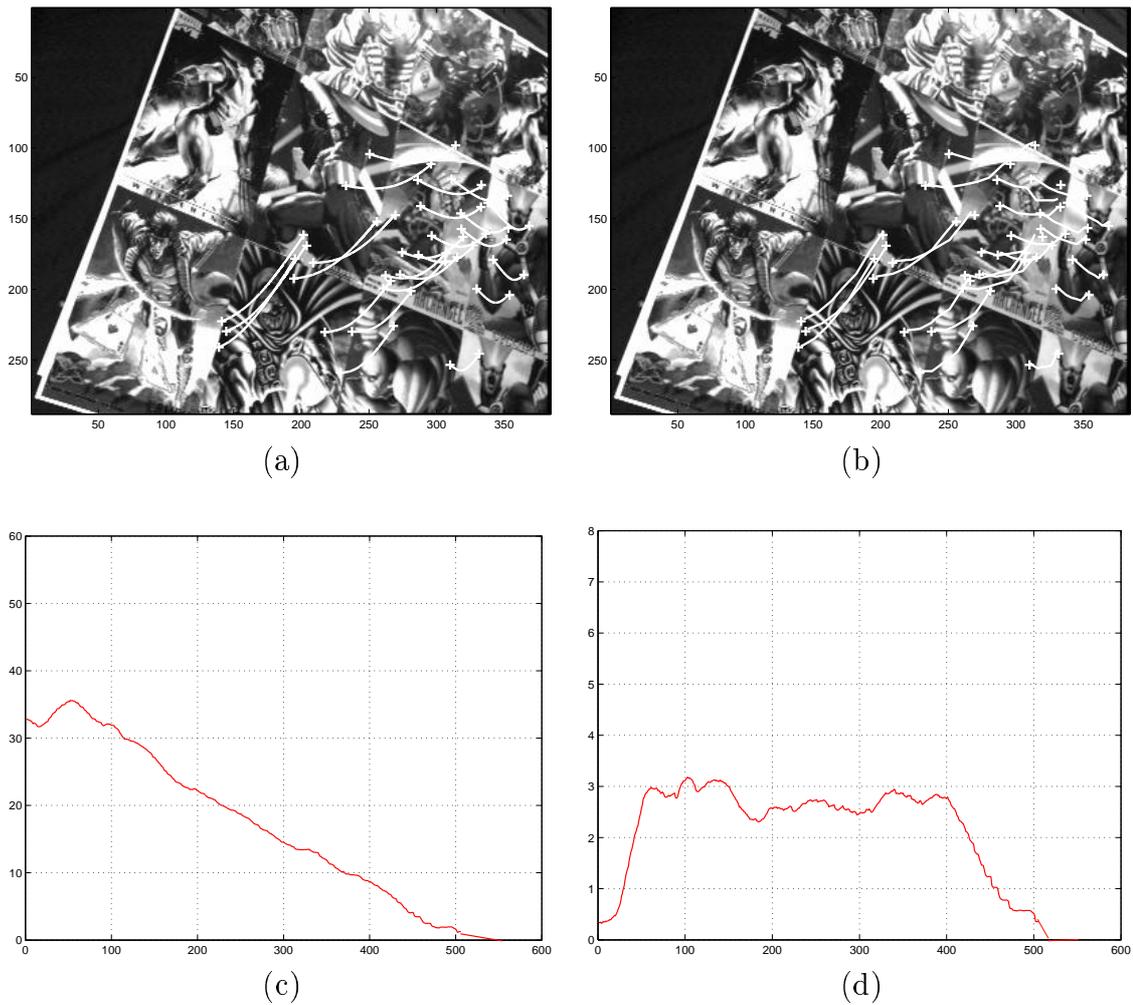


FIG. 5.39 – Mauvaise calibration (a) trajectoires planifiées, (b) trajectoires suivies, (c) erreurs par rapport aux coordonnées désirées dans l'image (pixels), (d) erreur de suivi (pixels)

5.2 Application à l'interpolation de N images

La réalisation d'un asservissement visuel nécessite au préalable la mise en correspondance de primitives géométriques communes aux images initiale et désirée. Lorsque la caméra observe une scène complexe, cette phase peut s'avérer très problématique si le déplacement de la caméra entre les prises de vues initiale et désirée est important, voire impossible si aucune primitive commune aux deux images n'est présente. Pour pallier cette difficulté, on peut utiliser un ensemble d'images intermédiaires. En effet, si deux images intermédiaires successives sont suffisamment proches, alors l'étape de mise en correspondance pourra s'effectuer aisément. Pour améliorer le comportement du système et construire une commande globale, il est alors nécessaire de construire la trajectoire des primitives visuelles dans l'image entre les images intermédiaires correspondant à une trajectoire réalisable de la caméra.

5.2.1 Le problème

Nous disposons d'un ensemble ordonné de $N + 1$ images acquises par la caméra $\mathcal{I} = \{I_0 \cdots I_N\}$ associé à un ensemble de $N + 1$ instants $\{t_0 \cdots t_N\}$ (se référer à la figure 5.40).

Nous supposons qu'entre deux images successives I_i et I_{i+1} , suffisamment de primitives visuelles puissent être extraites et mises en correspondance pour obtenir la colinéation \mathbf{G}_{ii+1} relative à un plan de référence Π . La matrice de colinéation $\mathbf{G}_{iN} = \mathbf{G}_i$ entre les images I_i et I_N peut simplement être obtenue en notant que :

$$\mathbf{G}_{iN} = \mathbf{G}_i = \mathbf{G}_{ii+1} \mathbf{G}_{i+1i+2} \cdots \mathbf{G}_{N-1N} \quad (5.49)$$

Nous disposons donc d'un ensemble ordonné de $N + 1$ matrices de colinéation $\mathcal{G} = \{\mathbf{G}_0 \cdots \mathbf{G}_N\}$ associé à un ensemble de $N + 1$ instants $\{t_0 \cdots t_N\}$. Nous désirons construire une trajectoire continue $\mathbf{G}(t)$, telle que $\mathbf{G}(t_i) = \mathbf{G}_i$ pour tout $i \in \{0 \cdots N\}$ et qui corresponde à une trajectoire minimale de la caméra.

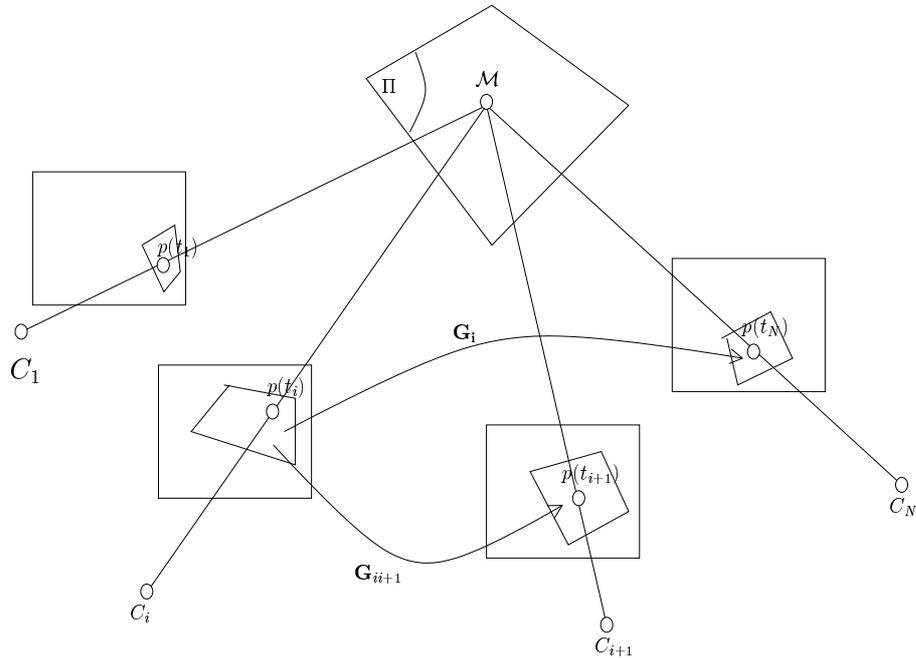


FIG. 5.40 – Interpolation de N images

5.2.2 Trajectoire minimale

Nous cherchons à déterminer la trajectoire de la matrice de colinéation entre \mathbf{G}_{i-1} et \mathbf{G}_i correspondant à un déplacement minimal de la caméra. Nous formulons ce problème *PCG1* de la même manière que *PC1*, seules les conditions aux limites sont différentes :

Trouver $\mathbf{G}(t)$ minimisant $J_5 = \int_{t_{i-1}}^{t_i} \mathbf{U}^T \mathbf{U} dt$
avec :

$$\mathbf{R}^T \dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times},$$

$$\mathbf{v} = \dot{\mathbf{t}}$$

$$\mathbf{G}(t) \propto \mathbf{K} \left(\mathbf{R} + \mathbf{t}_d \mathbf{n}^f \right) \mathbf{K}^{-1}$$

et les conditions aux limites suivantes :

$$\mathbf{G}(t_{i-1}) \propto \mathbf{G}_{i-1}$$

$$\mathbf{G}(t_i) \propto \mathbf{G}_i$$

Notons que les conditions aux limites du problème *PCG1* sont satisfaites si $\mathbf{R}(t_{i-1}) = \mathbf{R}_{i-1}$, $\mathbf{t}_{df}(t_{i-1}) = \mathbf{t}_{df_{i-1}}$, $\mathbf{R}(t_i) = \mathbf{R}_i$ et $\mathbf{t}_{df}(t_i) = \mathbf{t}_{df_i}$. La solution de *PCG1* peut être obtenue de la même manière que la solution *PC1*, et est donnée par :

$$\mathbf{G}(t) \propto (1 - \tau)\Phi_{i-1} + \tau\Phi_i + (\mathbf{G}_{i-1} - \Phi_{i-1})\Gamma(\boldsymbol{\theta}_i, \tau) \quad (5.50)$$

où $\tau = \frac{t-t_{i-1}}{t_i-t_{i-1}}$ et :

$$\begin{cases} \Gamma(\boldsymbol{\theta}_i, \tau) = \mathbf{K}e^{[\boldsymbol{\theta}_i]_{\times}\tau}\mathbf{K}^{-1} \\ \Phi_i = \mathbf{K}\mathbf{t}_{df_i}\mathbf{n}^{fT}\mathbf{K}^{-1} \end{cases} \quad (5.51)$$

avec $[\boldsymbol{\theta}_i]_{\times} = \log(\mathbf{R}_{i-1}^T\mathbf{R}_i)$. La trajectoire obtenue est indépendante des paramètres de la caméra si les matrices de colinéation à l'infini entre deux images successives sont connues. La trajectoire des primitives visuelles dans l'image et la loi de commande sont obtenues de la même manière que dans les paragraphes 5.1.4 et 5.1.6. Les trajectoires obtenues sont dérivables par morceaux. La vitesse de la caméra présente donc des discontinuités localisées aux passages entre deux images successives. Afin de palier cette difficulté, plusieurs approches peuvent être adoptées. La trajectoire obtenue peut par exemple être interpolée en utilisant des fonctions B-splines. La fonction de coût choisie peut également être modifiée afin d'assurer la dérivabilité de la trajectoire.

5.2.3 Résultats expérimentaux

Afin de réaliser une tâche de positionnement malgré la grande différence entre les images initiale et désirée (voir figures 5.41(a) et 5.41(b)), nous appliquons la méthode proposée précédemment. Dans le cas choisi, la phase de mise en correspondance de points d'intérêt est impossible. Nous utilisons donc l'ensemble d'images intermédiaires fourni par la figure 5.42 et obtenues automatiquement en utilisant des techniques d'indexation d'images [Mezouar 02]. Entre deux images successives, la mise en correspondance des points d'intérêt et le calcul de la matrice de colinéation s'effectuent aisément. Nous réalisons ensuite la phase de planification de trajectoires dans l'image. Les trajectoires obtenues sont illustrées sur la figure 5.43. La phase de suivi est ensuite réalisée. Les trajectoires dans l'image obtenues sont illustrées sur la figure 5.44. On observe, en comparant les figures 5.43 et 5.44, que les trajectoires suivies et planifiées sont similaires. La trajectoire de la caméra correspondante est fournie par la figure 5.45. L'erreur de suivi est donnée par la figure 5.46 et confirme l'efficacité du schéma proposé puisqu'elle reste faible (toujours inférieure à 5 pixels).

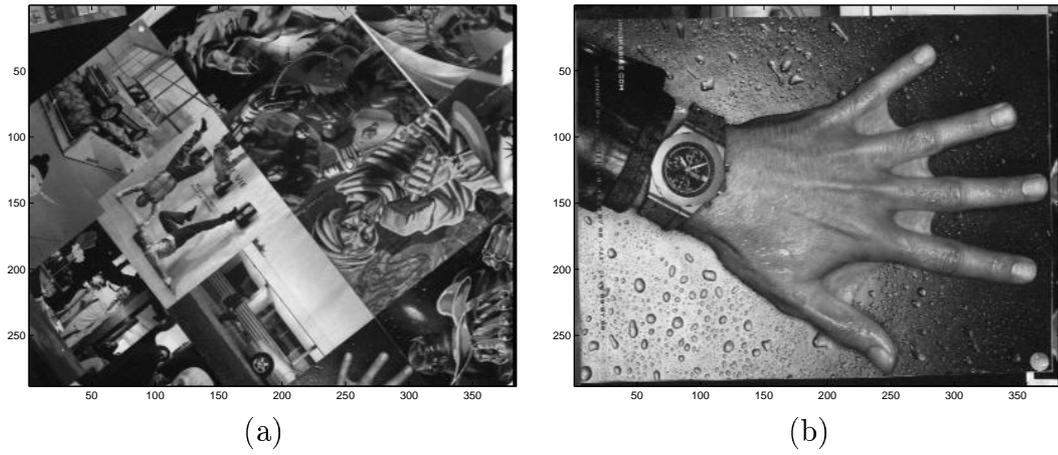


FIG. 5.41 – (a) Image initiale et (b) image désirée

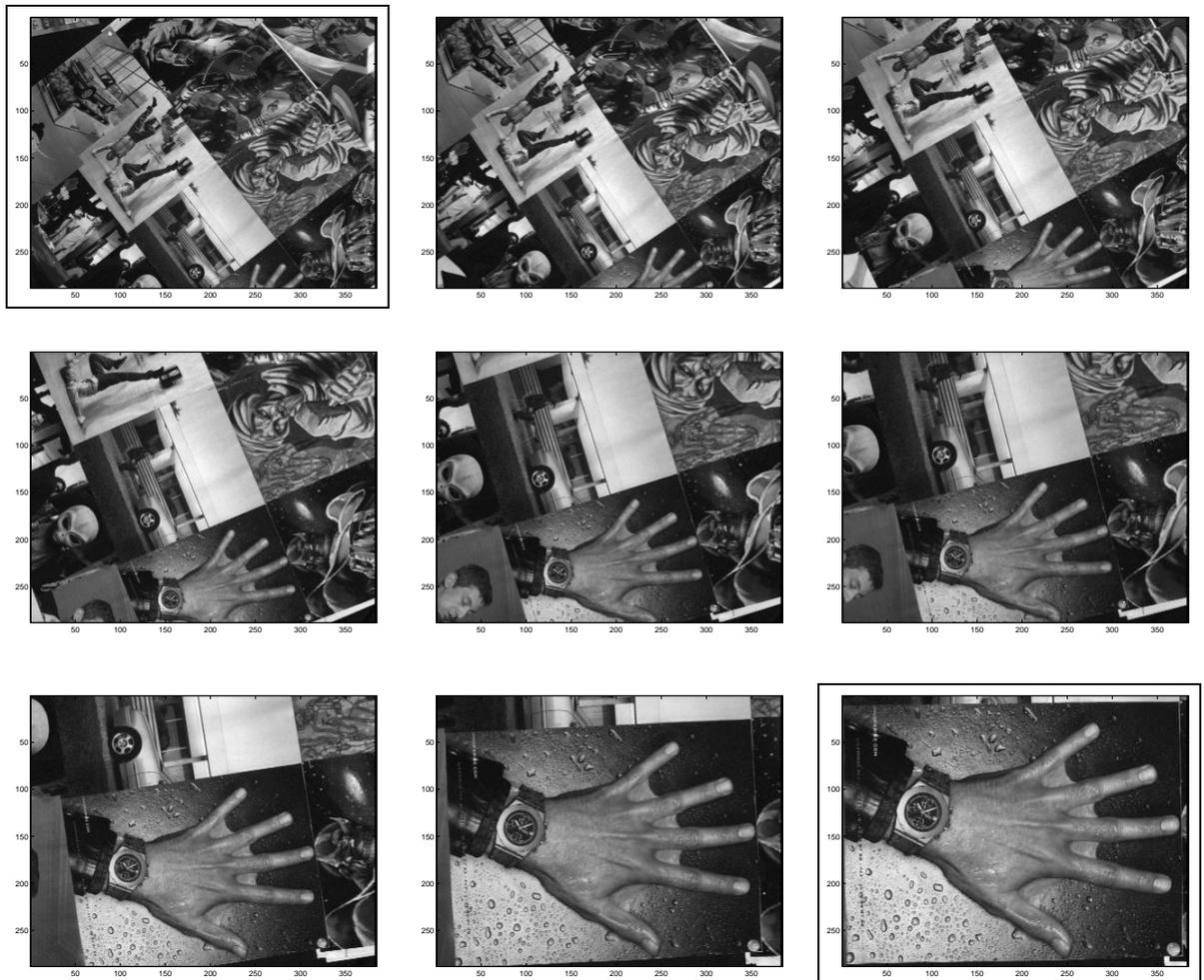
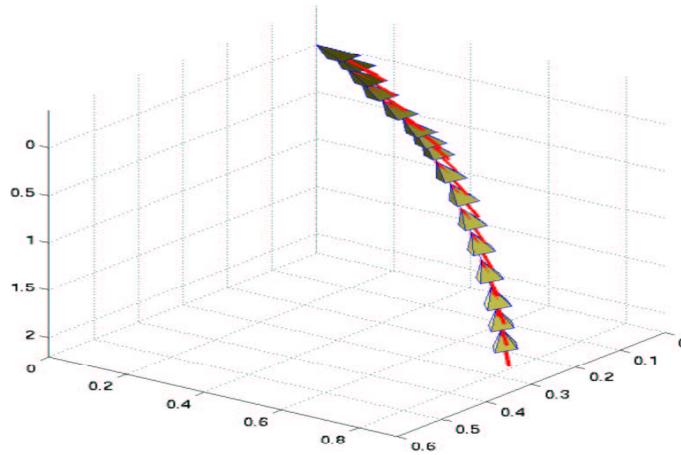
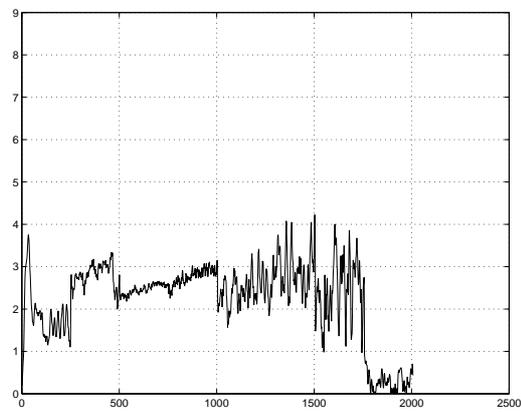




FIG. 5.43 – *Trajectoires planifiées dans l'image*



FIG. 5.44 – Trajectoires suivies dans l'image

FIG. 5.45 – *Trajectoire de la caméra*FIG. 5.46 – *Erreur de suivi*

5.3 Introduction de la contrainte de visibilité

Dans cette partie, à titre indicatif et afin de soulever les problèmes rencontrés lors de l'utilisation d'une méthode d'optimisation globale, nous mettons en œuvre la méthode de résolution numérique décrite dans l'annexe A. Celle-ci permet la prise en compte des contraintes de visibilité exprimées sous la forme de contraintes inégalités

5.3.1 Définition du problème

Nous supposons une nouvelle fois que la matrice de colinéation \mathbf{G}_0 liant les projections dans les images initiale et désirée des points appartenant à un plan de référence choisi, est disponible. A partir de \mathbf{G}_0 , nous estimons la matrice de rotation \mathbf{R}_0 , le vecteur de translation au facteur d'échelle d^f , \mathbf{t}_{df0} , et la normale au plan de référence exprimée dans le repère caméra final. Nous noterons $\boldsymbol{\theta} = \mathbf{u}\theta$, le produit de l'axe et de l'angle de rotation extrait de la matrice de rotation \mathbf{R} entre les positions courante et à atteindre de la caméra. Nous définissons l'état du système à l'instant t par le vecteur $\mathbf{x}(t) = [\mathbf{t}_{df}(t) \ \boldsymbol{\theta}(t)]$ et le vecteur de commande par :

$$\mathbf{U} = \dot{\mathbf{x}}(t) \quad (5.52)$$

Nous désirons déterminer la trajectoire de coût optimal de la caméra en présence de contraintes de visibilité. Nous formulons ce problème en définissant une fonction de coût qui est l'intégrale de la norme du vecteur de commande :

$$J = \int_0^1 \mathbf{U}^T \mathbf{U} dt \quad (5.53)$$

Nous voulons conserver l points dans le champ de vision de la caméra. Si on note \mathbf{p}_i , les coordonnées exprimées en pixel de la projection dans l'image d'un de ces l points, on désire que \mathbf{p}_i appartienne à $[u_{min} \ u_{max}] \times [v_{min} \ v_{max}]$; ce qui se traduit par $4 \times l$ contraintes inégalités :

$$\left\{ \begin{array}{l} u_i - u_{max} \leq 0 \quad , \quad i = 1 \dots l \\ u_{min} - u_i \leq 0 \quad , \quad i = 1 \dots l \\ v_i - v_{max} \leq 0 \quad , \quad i = 1 \dots l \\ v_{min} - v_i \leq 0 \quad , \quad i = 1 \dots l \end{array} \right.$$

Ces $4 \times l$ inégalités peuvent être réduites à $2 \times l$ contraintes de la forme :

$$\left\{ \begin{array}{l} in_{2i-2}(\mathbf{x}) = u_i^2 - a_u u_i + b_u \leq 0 \quad , \quad i = 1 \dots l \\ in_{2i-1}(\mathbf{x}) = v_i^2 - a_v v_i + b_v \leq 0 \quad , \quad i = 1 \dots l \end{array} \right. \quad (5.54)$$

avec $a_u = u_{max} + u_{min}$, $b_u = u_{max}u_{min}$, $a_v = v_{max} + v_{min}$ et $b_v = v_{max}v_{min}$. Afin de transformer ces contraintes inégalités en contraintes égalités, nous introduisons le vecteur de variable $\boldsymbol{\zeta} = [\zeta_1 \cdots \zeta_{n_{in}}]^T$ de dimension $2l$ (voir (A.32)) :

$$in_i(\mathbf{x}, \mathbf{u}) \leq 0 \Leftrightarrow eq_i(\mathbf{x}, \boldsymbol{\zeta}) = in_i(\mathbf{x}) + \zeta_i^2 = 0 \quad , \quad i = 0 \cdots 2l - 1 \quad (5.55)$$

Nous sommes maintenant en mesure de nous ramener à un problème de Bolza (voir paragraphe A.2.3). Pour cela, on définit l'Hamiltonien comme une fonction du vecteur d'état \mathbf{X} étendu de dimension $18 + 4l$ de la manière suivante (voir (A.33)) :

$$H(\mathbf{X}, \dot{\mathbf{X}}) = \mathbf{U}^T \mathbf{U} + \boldsymbol{\lambda}^T (\dot{\mathbf{x}} - \mathbf{U}) + \boldsymbol{\mu}^T \mathbf{eq}(\mathbf{x}, \boldsymbol{\zeta}) \quad (5.56)$$

où $\mathbf{eq} = [eq_0 \cdots eq_{2l-1}]$ et \mathbf{X} est le vecteur d'état étendu défini par (se référer à (A.34)) :

$$\mathbf{X} = [\mathbf{X}_x^T \quad \mathbf{X}_u^T \quad \mathbf{X}_\zeta^T \quad \mathbf{X}_\lambda^T \quad \mathbf{X}_\mu^T]^T \quad (5.57)$$

avec $\mathbf{X}_x = \mathbf{x}$, $\dot{\mathbf{X}}_u = \mathbf{U}$, $\dot{\mathbf{X}}_\zeta = [\zeta_1 \cdots \zeta_{2l}]$, $\dot{\mathbf{X}}_\lambda = [\lambda_1 \cdots \lambda_6]$, $\dot{\mathbf{X}}_\mu = [\mu_1 \cdots \mu_{2l}]$.

Les conditions aux limites sont données par (voir (A.36)) :

$$\mathbf{X}_x(t_0) = \mathbf{x}(t_0); \quad \mathbf{X}_u(t_0) = 0; \quad \mathbf{X}_\zeta(t_0) = 0; \quad \mathbf{X}_\lambda(t_0) = 0; \quad \mathbf{X}_\mu(t_0) = 0;$$

$$\mathbf{X}_x(t_f) = \mathbf{x}(t_f); \quad \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_u} \right) (t_f) = 0; \quad \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\zeta} \right) (t_f) = 0; \quad \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\lambda} \right) (t_f) = 0; \quad \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\mu} \right) (t_f) = 0.$$

avec :

$$\left(\frac{\partial H}{\partial \dot{\mathbf{X}}_u} \right) = 2\dot{\mathbf{X}}_u - \boldsymbol{\lambda}; \quad \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\zeta} \right) = 2[\mu_1 \zeta_1 \cdots \mu_{4l} \zeta_{4l}]^T; \quad \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\lambda} \right) = \dot{\mathbf{X}}_x - \dot{\mathbf{X}}_u; \quad \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\mu} \right) = \mathbf{eq}$$

Nous utilisons ensuite la méthode de résolution numérique décrite dans le paragraphe A.2.4 afin d'obtenir la trajectoire du vecteur d'état étendu. Une fois celle-ci obtenue, nous retrouvons aisément la trajectoire de la matrice de colinéation en utilisant la relation (5.21). La trajectoire de la matrice de colinéation est alors exploitée de la même manière que dans les paragraphes 5.1.4 pour construire la trajectoire des primitives visuelles dans l'image.

5.3.2 Exemple

L'objectif des expérimentations que nous présentons dans cette partie est la planification de la trajectoire de l'ensemble des pixels entre une image initiale et une image cible (images encadrées sur les figures 5.47 et 5.49).

Dans un premier temps, nous réalisons la planification sans introduire la contrainte de visibilité. Nous observons alors qu'une partie de l'objet d'intérêt quitte le champ

de vision de la caméra (à partir de l'instant $t=0.2$). Nous notons donc que la trajectoire rectiligne de la caméra (voir figure 5.48) ne permet pas de conserver entièrement l'objet d'intérêt dans l'image.

Nous utilisons alors la méthode décrite dans la section précédente (initialisée avec la solution obtenue lors de la première expérience) pour introduire la contrainte de visibilité. La trajectoire de la caméra dévie alors de la ligne droite (voir figure 5.50) et l'objet d'intérêt reste dans l'image (voir figure 5.49).

Nous ne développerons pas plus cette méthode, car elle nous semble mal adapté à notre problématique. En effet, cette méthode nécessite une initialisation. Cette étape peut s'avérer très délicate. Dans l'exemple que nous avons présenté, les trajectoires ont été initialisées en utilisant la solution obtenue lorsque les contraintes de visibilité ne sont pas prises en compte (ce qui correspond à une trajectoire selon une géodésique de la caméra). Dans les cas où la déviation (nécessaire au maintien de l'objet d'intérêt dans l'image) par rapport à cette initialisation est importante, une telle stratégie échoue. L'étape d'initialisation nécessite également une bonne calibration de la caméra afin d'obtenir les conditions aux limites à partir des images initiale et désirée.

En outre, la complexité de la mise en œuvre et le temps de calcul d'une telle méthode deviennent importants lorsque le nombre de contraintes (le nombre de points à conserver dans l'image) augmentent. Pour pallier ces difficultés, dans le chapitre suivant, nous allons utiliser une approche totalement différente (locale), basée sur la méthode des potentiels.

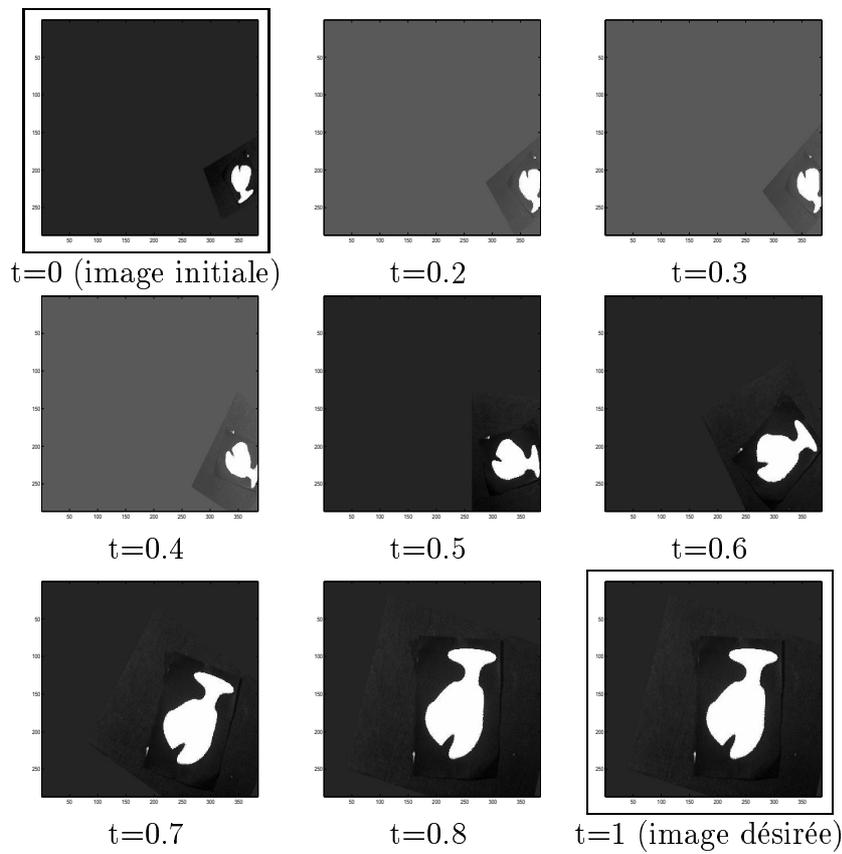


FIG. 5.47 – Trajectoire dans l'image : expérimentation sans contrainte de visibilité

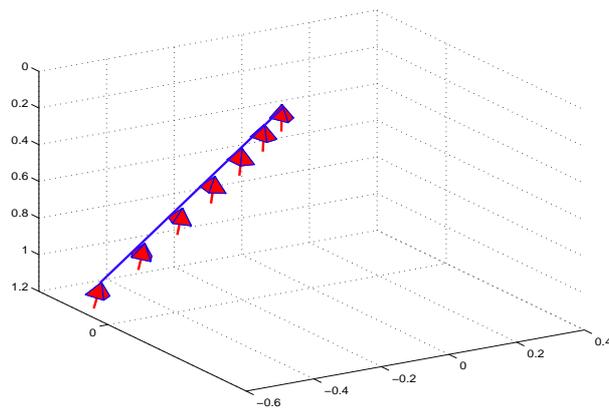


FIG. 5.48 – Trajectoire de la caméra : expérimentation sans contrainte de visibilité

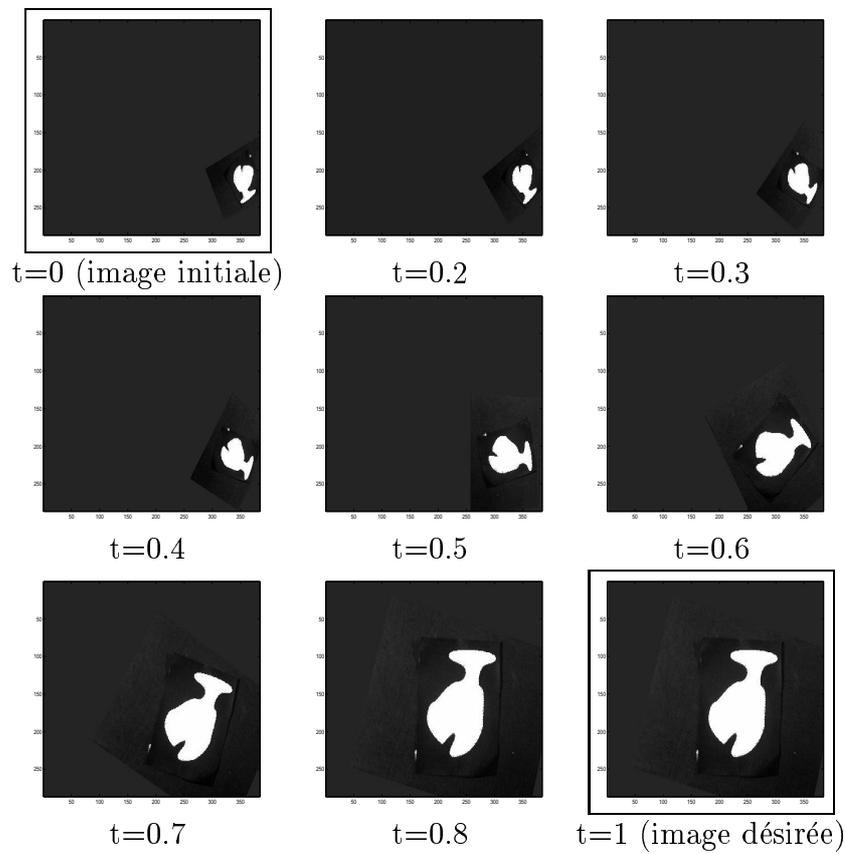


FIG. 5.49 – Trajectoire dans l'image : expérimentation avec contrainte de visibilité

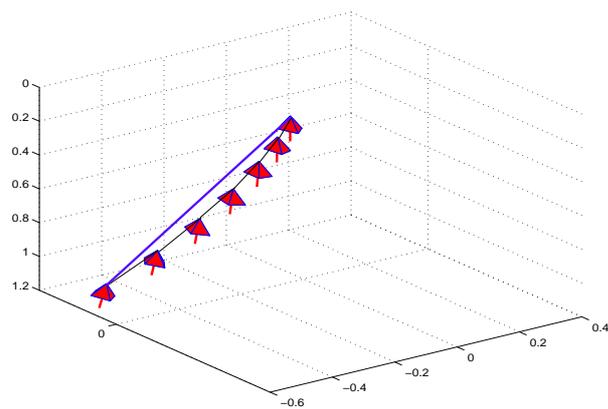


FIG. 5.50 – Trajectoire de la caméra : expérimentation avec contrainte de visibilité

5.4 Conclusion

Dans ce chapitre, nous avons vu comment, à partir de deux images d'une scène statique acquises par une caméra, la forme analytique de la trajectoire dans l'image de primitives visuelles pouvait être obtenue pour relier ces deux images. Les trajectoires dans l'image correspondent à un déplacement optimal (spatial et temporel) de la caméra dans l'espace de travail. Nous avons considéré les problèmes à minimum d'énergie et minimum d'accélération. Les solutions obtenues sont, comme attendu, des géodésiques où seule la paramétrisation temporelle change selon le critère d'optimalité retenu. D'autres types de critères, amenant à une paramétrisation spatiale et/ou temporelle différentes, pourraient éventuellement être choisis. Nous avons également montré que les trajectoires ainsi obtenues sont indépendantes de la calibration de la caméra. Nous avons ensuite utilisé un schéma de commande de type 2D pour suivre les trajectoires dans l'image. Nous avons également étudié le cas de l'interpolation d'un nombre quelconque d'images afin de pallier le problème de mise en correspondance des primitives visuelles lorsque les prises de vue initiale et finale sont distantes. En outre, les méthodes développées ne nécessitent pas les modèles des objets observés par la caméra. Nous avons appliqué les résultats obtenus à des tâches de positionnement, en utilisant une caméra CCD montée sur l'organe terminal d'un robot cartésien à six degrés de liberté.

Nous avons ensuite utilisé une méthode de résolution numérique des problèmes de commande optimale contraints afin d'obtenir une solution numérique quand la contrainte de visibilité est prise en compte. Malheureusement, même si cette approche fournit des résultats satisfaisants pour certaines configurations, celle-ci semble mal adaptée au problème posé par l'asservissement visuel, car elle nécessite, à la fois, une bonne initialisation et une calibration correcte de la caméra. D'autre part, sa mise en œuvre est lourde, en particulier, lorsque le nombre de contraintes devient important.

Pour pallier ces dernières difficultés, nous allons développer dans le chapitre suivant un schéma de planification basé sur la méthode des potentiels. Cette approche va nous permettre de prendre en compte simplement les contraintes de visibilité, mais également des contraintes liées au mécanisme (butées mécaniques).

Chapitre 6

Approche par la méthode des potentiels

Nous présentons dans ce chapitre une méthode de planification de primitives visuelles dans l'image permettant la prise en compte de contraintes liées au type de capteur utilisé et aux mécanismes du robot. Plus précisément, nous nous intéresserons aux contraintes de visibilité et d'évitement des butées articulaires du robot.

On considérera qu'on dispose des images initiale (I_i) et désirée (I_f) de la scène et du vecteur contenant les coordonnées articulaires du robot \mathbf{q}_i^* lorsque celui-ci est dans la configuration initiale (ce vecteur étant nécessaire uniquement lorsque les butées articulaires du robot sont prises en compte). On supposera que l'on peut extraire des deux images les vecteurs de dimension $2 \times n$ composés des coordonnées de n points \mathbf{p}_j de l'image $\mathbf{s}_i = [\mathbf{p}_i^{1^T} \cdots \mathbf{p}_i^{n^T}]^T$ et $\mathbf{s}_f = [\mathbf{p}_f^{1^T} \cdots \mathbf{p}_f^{n^T}]^T$. L'objectif est alors de fournir en sortie du planificateur, une trajectoire continue et dérivable $\mathbf{s}(t) = [\mathbf{p}_{(t)}^{1^T} \cdots \mathbf{p}_{(t)}^{n^T}]^T$ entre \mathbf{s}_i et \mathbf{s}_f . Pour cela, dans un premier temps, la trajectoire discrète de la caméra est obtenue et représentée par une série de N poses intermédiaires de la caméra dans \mathcal{W} , $\mathcal{X} = \{\mathbf{x}_k / k = 1 \cdots N\}$. Ensuite, la trajectoire discrétisée des primitives sans l'image $\mathcal{S} = \{\mathbf{s}_k / k = 1 \cdots N\}$ et du robot dans l'espace articulaire $\mathcal{Q} = \{\mathbf{q}_k / k = 1 \cdots N\}$ est obtenue à partir de \mathcal{X} . Enfin, on obtiendra une trajectoire géométrique continue et dérivable des primitives dans l'image, auxquelles on associe une variation temporelle $\mathbf{s}(t)$ qui sera utilisée comme consigne d'un asservissement visuel 2D.

Après une description de la méthode classique des potentiels, nous verrons comment modifier cette méthode afin de tenir compte de contraintes relatives à des espaces différents (espace articulaire, espace de travail, espace de tâche). La seconde partie de ce chapitre sera consacrée à l'étude des cas où le modèle de la cible est disponible. Nous étendrons ensuite la méthode proposée au problème de planification quand le modèle de la cible n'est pas disponible. Dans ce cas, nous verrons qu'il est possible d'obtenir des résultats concernant la robustesse du planificateur vis-à-vis

d'éventuelles erreurs sur la calibration de la caméra et sur la profondeur d^f . Dans la quatrième partie de ce chapitre, nous utilisons une interpolation par des fonctions B-splines afin de construire des trajectoires dans l'image continues et dérivables. Nous utilisons ensuite un schéma d'asservissement visuel 2D pour suivre les trajectoires obtenues. La dernière partie sera consacrée aux résultats expérimentaux obtenus sur la plate-forme robotique de l'IRISA.

6.1 La méthode des potentiels

Le principe de la génération de trajectoire proposé dans ce chapitre est inspiré de la méthode des fonctions de potentiel. Cette méthode a été développée à l'origine pour la planification de trajectoire dans un espace de travail encombré [Khatib 86], [Latombe 91].

6.1.1 L'approche classique

Dans cette approche, la situation du robot est représentée à chaque instant par un vecteur \mathbf{x} dans \mathcal{W} . L'espace de travail du robot est baigné dans un champ de potentiel artificiel V , dont les variations locales doivent refléter la structure géométrique. Les mouvements du robot sont alors sous l'influence du champ de potentiel artificiel V :

$$\begin{cases} V : \mathcal{W} \rightarrow \mathfrak{R} \\ \mathbf{x} \rightarrow V(\mathbf{x}) \end{cases} \quad (6.1)$$

Le potentiel artificiel peut être décomposé en la somme d'un potentiel attractif V_a agissant sur le robot de manière à l'amener dans la configuration désirée, et d'un potentiel répulsif V_r permettant de conserver le robot dans son espace libre \mathcal{W}_{free} :

$$V(\mathbf{x}) = V_a(\mathbf{x}) + V_r(\mathbf{x}) \quad (6.2)$$

Le potentiel attractif est défini comme une fonction indépendante de l'ensemble des configurations répulsives. Il doit atteindre son minimum lorsque le robot est dans la configuration désirée. Le potentiel répulsif est choisi indépendamment de la configuration désirée, décroissant quand le robot s'éloigne des configurations indésirables et prenant une très grande valeur quand il est dans le voisinage d'une d'entre elles. La planification des mouvements du robot est réalisée de manière itérative. A chaque itération, la force artificielle $\mathbf{F}(\mathbf{x})$ est induite par le champ de potentiel V dans son état courant. Cette force artificielle est définie comme le vecteur transposé du gradient du potentiel V par rapport au vecteur \mathbf{x} :

$$\begin{cases} \mathbf{F} : \mathcal{TW} \rightarrow \mathcal{TW} \\ \mathbf{x} \rightarrow -\vec{\nabla}^T V(\mathbf{x}) \end{cases} \quad (6.3)$$

où $\vec{\nabla}$ représente l'opérateur gradient. La force artificielle peut donc, comme le potentiel V , être décomposée en la somme d'une force attractive $\mathbf{F}_a(\mathbf{x})$ et d'une force répulsive $\mathbf{F}_r(\mathbf{x})$, avec :

$$\mathbf{F}_a(\mathbf{x}) = -\vec{\nabla}_{\mathbf{x}}^T V_a$$

$$\mathbf{F}_r(\mathbf{x}) = -\vec{\nabla}_{\mathbf{x}}^T V_r$$

La trajectoire est alors obtenue en suivant la direction courante fournie par \mathbf{F} , vue comme la direction la mieux adaptée pour atteindre la configuration courante, en respectant l'encombrement de l'espace de travail du robot. La trajectoire du robot dans \mathcal{W} peut alors être construite comme l'union de segments successifs en partant de la configuration initiale \mathbf{x}_i . Le segment courant est orienté dans la direction opposée au gradient de la fonction de potentiel dans l'état qu'elle a atteint à l'étape précédente. La trajectoire discrète du robot est alors donnée par l'équation de transition suivante :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \varepsilon_k \frac{\mathbf{F}(\mathbf{x}_k)}{\|\mathbf{F}(\mathbf{x}_k)\|} \quad (6.4)$$

où k est l'index d'incrément et ε_k est un réel positif représentant la longueur du segment k . En théorie, $\|\mathbf{F}(\mathbf{x}_k)\|$ peut être nulle lorsque la fonction de potentiel atteint un minimum (local ou global). Nous verrons dans la suite que les minimums locaux ne posent pas de problèmes en pratique. En ce qui concerne le minimum global celui-ci est seulement approché car la phase de planification de trajectoires est stoppée lorsque la distance entre les points courants et désirés dans l'image est inférieure à une distance non nulle choisie par l'utilisateur.

6.1.2 L'approche utilisée

Nous utiliserons une approche un peu différente de celle décrite précédemment afin d'intégrer efficacement des contraintes relatives à des espaces différents. Pour cela, considérons le problème d'optimisation sans contrainte suivant :

$$\min V(\mathbf{x}), \quad \mathbf{x} \in \mathbf{R}^p \quad (6.5)$$

Une méthode numérique classique pour obtenir le minimum de V consiste à faire varier $\mathbf{x}(t)$ selon l'équation d'évolution suivante :

$$\dot{\mathbf{x}} = -\epsilon \mathbf{Q} \vec{\nabla}_{\mathbf{x}}^T V \quad (6.6)$$

où ϵ est un réel positif et \mathbf{Q} une matrice définie positive et constante. En prémultipliant chaque membre de l'équation précédente par $\vec{\nabla}_{\mathbf{x}} V$, on obtient :

$$\frac{d}{dt} V(\mathbf{x}) = -\epsilon \vec{\nabla}_{\mathbf{x}} V \mathbf{Q} \vec{\nabla}_{\mathbf{x}}^T V \leq 0 \quad (6.7)$$

La fonction V décroît au cours du temps tant que $\vec{\nabla}_{\mathbf{x}}V$ est non nul, et est constant quand $\vec{\nabla}_{\mathbf{x}}V$ est nul. Un choix classique et simple pour la matrice \mathbf{Q} est de la prendre égale à la matrice identité. Dans ce cas, \mathbf{x} évolue dans la direction opposée au gradient de V par rapport à \mathbf{x} . Cette stratégie est adoptée dans l'approche classique de planification de trajectoire par fonction de potentiel décrite précédemment, où $\mathbf{F} = -\vec{\nabla}_{\mathbf{x}}^T V$. Considérons maintenant une fonction de potentiel $V_{\mathbf{f}} = V(\mathbf{f}(\mathbf{x}))$, où \mathbf{f} à valeur dans \mathbf{R}^n est dérivable sur tout l'espace de travail \mathcal{W} . L'équation d'évolution de \mathbf{f} , quand les variations de \mathbf{x} vérifient la relation (6.6), est donnée par :

$$\dot{\mathbf{f}} = -\epsilon \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \mathbf{Q} \vec{\nabla}_{\mathbf{x}}^T V = -\epsilon \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \mathbf{Q} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T \vec{\nabla}_{\mathbf{f}}^T V \quad (6.8)$$

Pour que \mathbf{f} se déplace dans la direction opposée au gradient de V par rapport à \mathbf{f} , la matrice \mathbf{Q} doit être choisie de manière adéquate, par exemple :

$$\mathbf{Q} = \mathbf{Q}_{\mathbf{f}} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^+ \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{+T} \quad (6.9)$$

Dans ce cas, la matrice \mathbf{Q} est positive et la relation (6.7) est donc vérifiée. L'équation (6.8) donnant l'évolution de \mathbf{f} peut alors s'écrire :

$$\dot{\mathbf{f}} = -\epsilon \vec{\nabla}_{\mathbf{f}}^T V \quad (6.10)$$

En utilisant les conventions standard utilisées dans la méthode des potentiels, la force artificielle associée au champ de potentiel $V(\mathbf{f}(\mathbf{x}))$ est donnée par :

$$\mathbf{F}_{\mathbf{f}}(\mathbf{x}) = -\mathbf{Q} \vec{\nabla}_{\mathbf{x}}^T V_{\mathbf{f}} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^+ \vec{\nabla}_{\mathbf{f}}^T V \quad (6.11)$$

Si l'on considère plusieurs fonctions de potentiel, la force artificielle dominante dérivée du potentiel $V_{\mathbf{f}}$ crée un mouvement dominant de \mathbf{f} dans la direction opposée au gradient de $V_{\mathbf{f}}$ par rapport à \mathbf{f} . En pratique, en utilisant ce choix pour la matrice \mathbf{Q} , il sera plus simple de contrôler l'influence relative de chacune des forces artificielles et donc de maîtriser la trajectoire de la caméra dans l'espace de travail et de l'objet d'intérêt dans l'image.

Dans notre cas, notre objectif est de transférer le système d'un état initial à un état désiré dans l'espace de tâche (par conséquent dans l'image) en satisfaisant les contraintes suivantes :

1. les trajectoires des primitives dans l'image correspondent à des mouvements physiquement réalisables du robot;
2. toutes les primitives images considérées doivent rester dans le champ de vision de la caméra;
3. le robot doit respecter les limitations mécaniques (butées articulaires);

Pour que la première contrainte soit satisfaite, les mouvements de la caméra sont dans un premier temps planifiés dans l'espace de travail \mathcal{W} et ensuite les positions successives des primitives dans l'espace de tâche sont obtenues par simple projection. Dans cette optique, le potentiel attractif ($V_{\mathbf{x}}$) amenant le robot vers sa position désirée (\mathbf{x}_f) est défini dans \mathcal{W} . Les deux autres contraintes sont introduites à travers deux potentiels répulsifs. La fonction de potentiel liée aux contraintes de visibilité est directement définie dans l'image et est notée $V_{\mathbf{s}}$. La fonction de potentiel liée aux contraintes d'évitement des butées articulaires est définie dans l'espace articulaire du robot et est notée $V_{\mathbf{q}}$. La force totale est donnée par :

$$\mathbf{F} = \mathbf{F}_{\mathbf{x}} + \gamma \mathbf{F}_{\mathbf{s}} + \chi \mathbf{F}_{\mathbf{q}} \quad (6.12)$$

où γ et χ sont deux réels positifs permettant d'ajuster l'influence relative des différentes forces. La force totale donnée par l'équation (6.12) peut potentiellement conduire à un minimum local de la fonction de potentiel. En pratique, nous n'avons jamais rencontré de telles configurations et les paramètres γ et χ ont été fixés à 1. Notons que les six composantes de la force totale (la somme pondérée des trois forces) doivent s'annuler au même instant pour atteindre un minimum local. Une telle configuration a une faible probabilité d'être atteinte. Dans le cas où un minimum local de la fonction de potentiel est atteint, une stratégie consistant à exécuter un mouvement en favorisant les forces répulsives (c'est à dire en augmentant γ et χ) peut être utilisée pour sortir de cette configuration. Evidemment, une telle stratégie ne garantit pas formellement la convergence vers le minimum global de la fonction de potentiel. Pour garantir cette propriété, des fonctions de potentiel sans minimum local stable peuvent être construites (fonctions de navigation) [Cowan 99]. Cependant, la construction de ces fonctions nécessite une connaissance parfaite de la topologie de l'espace de travail du robot. De nombreux avantages des méthodes proposées dans ce manuscrit ne sont donc pas conservés (robustesse vis-à-vis des erreurs de modélisation, application à des objets dont les modèles CAO sont inconnus) si de telles fonctions sont utilisées.

D'après la relation (6.11), les forces artificielles peuvent s'écrire de la façon suivante :

$$\left\{ \begin{array}{l} \mathbf{F}_{\mathbf{x}} = - \left(\frac{\partial \mathbf{x}}{\partial \mathbf{x}} \right)^+ \vec{\nabla}_{\mathbf{x}}^T V_{\mathbf{x}} = - \vec{\nabla}_{\mathbf{x}}^T V_{\mathbf{x}} \\ \mathbf{F}_{\mathbf{s}} = - \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right)^+ \vec{\nabla}_{\mathbf{s}}^T V_{\mathbf{s}} = - \mathbf{M}^+ \mathbf{L}^+ \vec{\nabla}_{\mathbf{s}}^T V_{\mathbf{s}} \\ \mathbf{F}_{\mathbf{q}} = - \left(\frac{\partial \mathbf{q}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right)^+ \vec{\nabla}_{\mathbf{q}}^T V_{\mathbf{q}} = - \mathbf{M}^+ \mathbf{J}(\mathbf{q}) \vec{\nabla}_{\mathbf{q}}^T V_{\mathbf{q}} \end{array} \right. \quad (6.13)$$

où \mathbf{M} est une matrice jacobienne qui lie les variations de la vitesse de la caméra $\boldsymbol{\tau}_c$ aux variations de la paramétrisation choisie pour représenter l'attitude de la caméra dans \mathcal{W} : $\boldsymbol{\tau}_c = \mathbf{M}\dot{\mathbf{x}}$. La forme de cette matrice sera donnée dans la suite du document

pour la paramétrisation choisie. La matrice $\mathbf{J}(\mathbf{q})$ représente le jacobien du robot et la matrice \mathbf{L} est la matrice d'interaction relative au vecteur des primitives visuelles \mathbf{s} .

6.2 Cas d'une cible connue

Dans cette partie, les paramètres intrinsèques de la caméra seront supposés connus. On considérera également qu'on dispose d'un modèle de la cible observée sous la forme des coordonnées \mathbf{M}_o^j d'un ensemble de n points \mathcal{M}^j dans un repère qui lui est associé, \mathcal{F}_o . Si l'on dispose d'au moins quatre de ces points, alors il est possible à partir de leur projection dans l'image de calculer les poses initiale et désirée de la caméra dans \mathcal{F}_o [Dementhon 95, Lowe 91]. Plus précisément, on obtient aisément les matrices de rotation ${}^i\mathbf{R}_o$ et ${}^f\mathbf{R}_o$ et les vecteurs de translation ${}^i\mathbf{t}_o$ et ${}^f\mathbf{t}_o$ entre les repères \mathcal{F}_i et \mathcal{F}_o , et \mathcal{F}_f et \mathcal{F}_o , respectivement.

Dans la suite du document, nous utiliserons l'index k pour signifier que l'on considère la situation d'un paramètre à l'instant $k\Delta T$ où ΔT est une période d'échantillonnage choisie. Pour simplifier l'exposé, nous utiliserons simplement l'index k en lieu et place de l'index ck . Par exemple, le repère lié à la caméra à l'instant k sera noté \mathcal{F}_k plutôt que \mathcal{F}_{ck} .

6.2.1 Trajectoire de la caméra dans \mathcal{W}

Soit ${}^f\mathbf{R}_k$ et ${}^f\mathbf{t}_k$ la matrice de rotation et le vecteur de translation entre \mathcal{F}_k et \mathcal{F}_f (voir figure 6.1), nous choisissons comme paramétrisation de l'attitude de la caméra à l'instant k , le vecteur $\mathbf{x}_k = [{}^f\mathbf{t}_k^T \boldsymbol{\theta}_k]$ avec $\boldsymbol{\theta}_k = (\mathbf{u}\theta)_k$, où \mathbf{u}_k et θ_k sont l'axe unitaire et l'angle obtenus à partir de la rotation ${}^f\mathbf{R}_k$. Nous avons donc $\mathbf{x}_i = [{}^f\mathbf{t}_i^T \boldsymbol{\theta}_i]^T$ quand la caméra est dans sa position initiale, et $\mathbf{x}_f = \mathbf{0}_{1 \times 6}$ quand la caméra est dans sa position désirée. L'attitude initiale de la caméra \mathbf{x}_i est obtenue à partir de ${}^f\mathbf{t}_i$ et ${}^f\mathbf{R}_i$, avec :

$$\begin{cases} {}^f\mathbf{R}_i &= {}^f\mathbf{R}_o {}^i\mathbf{R}_o^T \\ {}^f\mathbf{t}_i &= -{}^f\mathbf{R}_i {}^i\mathbf{t}_o + {}^f\mathbf{t}_o \end{cases} \quad (6.14)$$

En utilisant l'équation de transition (6.4), où les forces répulsives seront données par la suite (voir les Sections 6.2.4, 6.2.5 et 6.2.6), la trajectoire de la caméra $\mathcal{X} = \{\mathbf{x}_k / k = 1 \cdots N\}$ dans \mathcal{W} est construite comme une séquence de N segments dont le point de départ est \mathbf{x}_i . En outre, la profondeur Z_k^j de chaque point \mathcal{M}^j dans le repère \mathcal{F}_k peut également être aisément obtenue, avec :

$$\widetilde{\mathbf{M}}_k^j = [X_k^j Y_k^j Z_k^j]^T = [{}^f\mathbf{R}_k^T {}^f\mathbf{R}_o \quad {}^f\mathbf{R}_k^T ({}^f\mathbf{t}_o - {}^f\mathbf{t}_k)] \mathbf{M}_o^j \quad (6.15)$$

La profondeur Z_k^j de chaque point \mathcal{M}^j est très importante car elle apparaît explicitement à la fois, dans l'expression des forces répulsives, et dans la loi de commande.

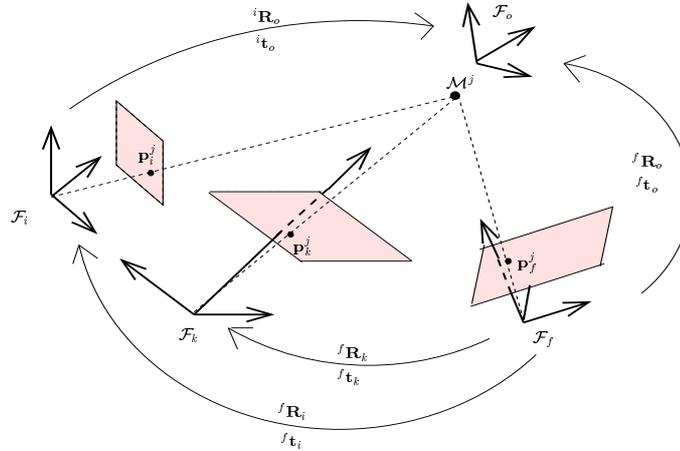


FIG. 6.1 – Trajectoire de la caméra dans l'espace de travail

Un chemin complet de la caméra entre sa position initiale et désirée est donc obtenu sous la forme d'une séquence \mathcal{X} de N poses intermédiaires. Un schéma de commande du type asservissement visuel $3D$ pourrait donc être utilisé afin de contrôler les mouvements de la caméra sur cette trajectoire. Cependant, il nous a semblé préférable de construire les trajectoires des primitives dans l'image, afin d'exploiter au mieux la stabilité et la robustesse aux erreurs de modélisation et aux bruits de mesure qu'offrent les asservissements visuels $2D$. En outre, cela nous permet, d'éviter l'estimation de la pose de la caméra à chaque itération de la boucle d'asservissement.

6.2.2 Trajectoire des primitives dans l'image

Pour réaliser un suivi de trajectoire directement dans l'espace du capteur, il est nécessaire de construire la trajectoire des primitives choisies dans l'image. Les coordonnées exprimées en pixel \mathbf{p}_k^j de la projection dans l'image à l'instant k de chaque point \mathcal{M}^j , dont les coordonnées \mathbf{M}_o^j dans le repère de l'objet \mathcal{F}_o sont parfaitement connues, sont obtenues en utilisant l'hypothèse classique que la caméra réalise une projection perspective parfaite avec pour centre, le centre optique de la caméra (voir (2.9)) :

$$\alpha_k^j \mathbf{p}_k^j = [\alpha_k^j u_k^j \quad \alpha_k^j v_k^j \quad \alpha_k^j]^T = \mathbf{K} [{}^k\mathbf{R}_o \quad {}^k\mathbf{t}_o] \mathbf{M}_o^j \quad (6.16)$$

\mathbf{p}_k^j est alors aisément obtenu à partir de (6.16) en divisant $\alpha_k^j \mathbf{p}_k^j$ par sa dernière coordonnée.

6.2.3 Trajectoire dans l'espace articulaire du robot

Pour anticiper une possible rencontre d'un des axes du robot avec une éventuelle butée mécanique et ainsi pouvoir l'éviter, il est nécessaire d'avoir une estimation de la

trajectoire des axes du robot dans l'espace articulaire. En effet, l'estimation courante de l'état des articulations du robot sera utilisée pour le calcul de la force répulsive associée à l'évitement des butées articulaires. Plusieurs méthodes sont possibles pour estimer les coordonnées articulaires à partir de la trajectoire de la caméra mais toutes ont le point commun de nécessiter une bonne estimation de la trajectoire cartésienne de la caméra. Ces méthodes nécessitent également la connaissance des coordonnées articulaires du robot à l'instant initiale \mathbf{q}_0 qui dans notre cas seront obtenues en même temps que l'acquisition de l'image initiale.

Représentons la situation du robot dans son espace articulaire par son vecteur de coordonnées articulaires $\mathbf{q} = [q^1 \cdots q^m]^T$ (dénomé \mathbf{q}_k à l'itération k), alors une première méthode consiste à utiliser le modèle géométrique inverse du robot (**MGI**):

$$\mathbf{q}_k = \text{MGI}(\mathbf{x}_k)$$

On peut également utiliser le jacobien du robot $\mathbf{J}(\mathbf{q})$. En effet, on a :

$$\frac{\partial \mathbf{q}}{\partial \mathbf{x}} = \frac{\partial \mathbf{q}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \mathbf{J}^+(\mathbf{q})\mathbf{M} \quad (6.17)$$

où \mathbf{M} est une matrice Jacobienne de dimension 6×6 qui lie les variations de la vitesse de la caméra $\boldsymbol{\tau}_c$, et les variations de \mathbf{x} . Cette matrice est donnée par :

$$\mathbf{M} = \begin{bmatrix} {}^f\mathbf{R}_k^T & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{L}_{wk}^+ \end{bmatrix} \quad (6.18)$$

où la matrice \mathbf{L}_{wk}^+ est donnée par [Malis 99]:

$$\mathbf{L}_{wk}^+ = \mathbf{I} + \frac{\theta_k}{2} \text{sinc}^2\left(\frac{\theta_k}{2}\right) [\mathbf{u}_k]_{\times} + (1 - \text{sinc}(\theta_k)) [\mathbf{u}_k]_{\times}^2 \quad (6.19)$$

Le vecteur des coordonnées articulaires du robot peut alors être obtenu à partir de \mathbf{x} , en effectuant le développement d'ordre 1 autour de \mathbf{q}_k de la relation (6.17):

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^+(\mathbf{q}_k) \mathbf{M} (\mathbf{x}_{k+1} - \mathbf{x}_k) \quad (6.20)$$

Nous choisirons cette dernière méthode qui conduit à des résultats satisfaisants, une étude comparative des différentes méthodes reste cependant à effectuer. Nous verrons dans la partie consacrée aux résultats expérimentaux les limitations d'une telle approche.

Dans la partie suivante, nous allons décrire les différentes fonctions de potentiel que nous avons choisies pour amener le robot dans sa configuration désirée le long d'une trajectoire respectant les contraintes que nous nous sommes fixées.

6.2.4 Champ de potentiel attractif

Le champ de potentiel attractif $V_{\mathbf{x}}$ est simplement défini comme une fonction parabolique de façon à minimiser la distance entre la position courante et la position désirée de la caméra :

$$V_{\mathbf{x}}(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathbf{x}_f\|^2 = \frac{1}{2} \|\mathbf{x}\|^2 \quad (6.21)$$

La fonction $V_{\mathbf{x}}$ est positive ou nulle et atteint son minimum en \mathbf{x}_f où $V_{\mathbf{x}}(\mathbf{x}_f) = 0$. Elle génère une force $\mathbf{F}_{\mathbf{x}}$ qui converge linéairement vers la configuration désirée

$$\mathbf{F}_{\mathbf{x}}(\mathbf{x}) = -\vec{\nabla}_{\mathbf{x}}^T V_{\mathbf{x}} = -\mathbf{x} \quad (6.22)$$

Quand les fonctions de potentiel répulsives ne sont pas nécessaires, l'équation de transition (6.4) s'écrit :

$$\mathbf{x}_{k+1} = \left(1 - \frac{\varepsilon_k}{\|\mathbf{x}_k\|}\right) \mathbf{x}_k \quad (6.23)$$

Toutes les positions \mathbf{x}_k appartiennent donc à la droite passant par \mathbf{x}_i et \mathbf{x}_f . Il en résulte que le centre de projection de la caméra effectue effectivement une ligne droite dans l'espace de travail \mathcal{W} , puisque le vecteur paramétrisant l'attitude de la caméra est défini dans un repère fixe (i.e. \mathcal{F}_f). Cependant, en suivant cette trajectoire, l'objet d'intérêt peut sortir du champ de vision de la caméra, ou un ou plusieurs axes du robot peuvent atteindre leurs butées articulaires. Afin d'éviter ces problèmes potentiels, des forces répulsives sont introduites et modifient la trajectoire du robot quand cela est nécessaire.

6.2.5 Champ répulsif relatif aux butées articulaires

Une configuration articulaire \mathbf{q} du robot est dite acceptable si chacune de ses composantes est suffisamment éloignée de la butée articulaire correspondante. Plus précisément, la configuration \mathbf{q} est acceptable si pour tout $j = 1 \dots m$, $q^j \in [q_{min}^j + l^j; q_{max}^j - l^j]$, où q_{min}^j et q_{max}^j représentent les valeurs minimale et maximale que l'axe j peut atteindre, et l^j représente la distance d'influence de la butée mécanique j . On appellera \mathcal{L} le sous-ensemble de l'espace articulaire des configurations acceptables. Le champ de potentiel répulsif $V_{\mathbf{q}}$ est alors défini de la façon suivante (voir figure 6.2) :

$$V_{\mathbf{q}}(\mathbf{q}) = \begin{cases} -v_q^2 \log \left(\prod_{j=1}^m \left(1 - \frac{q^j}{q_{max}^j}\right) \left(\frac{q^j}{q_{min}^j} - 1\right) \right) & \text{si } \mathbf{q} \notin \mathcal{L} \\ 0 & \text{si } \mathbf{q} \in \mathcal{L} \end{cases} \quad (6.24)$$

Pour obtenir un champ de potentiel $V_{\mathbf{q}}$ continu et dérivable, la fonction v_q est choisie

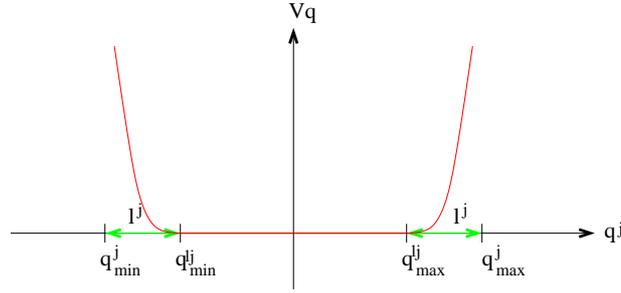


FIG. 6.2 – Potentiel répulsif associé à l'évitement des butées articulaires

comme une fonction bornée nulle sur la frontière de \mathcal{L} :

$$v_q(\mathbf{q}) = \prod_{j=1}^n (q_{max}^{lj} - q^j)(q^j - q_{min}^{lj}) \quad (6.25)$$

où $q_{min}^{lj} = q_{min}^j + l^j$ et $q_{max}^{lj} = q_{max}^j - l^j$. La fonction $V_{\mathbf{q}}$ est positive ou nulle, tend vers l'infini quand une des composantes de \mathbf{q} se rapproche d'une butée articulaire, et est nulle quand la distance entre chaque composante de \mathbf{q}^j et la butée mécanique correspondante est supérieure à l^j . La force artificielle $\mathbf{F}_{\mathbf{q}}$ qui dérive du champ de potentiel $V_{\mathbf{q}}$ est alors (voir l'équation (6.13)) :

$$\mathbf{F}_{\mathbf{q}} = -\mathbf{M}^+ \mathbf{J} \vec{\nabla}_{\mathbf{q}}^T V_{\mathbf{q}} \quad (6.26)$$

où :

$$\vec{\nabla}_{\mathbf{q}}^T V_{\mathbf{q}} = \begin{cases} 2 \begin{bmatrix} \nabla v_q^1 \\ \vdots \\ \nabla v_q^i \\ \vdots \\ \nabla v_q^m \end{bmatrix} v_q \phi_{\mathbf{q}} + v_q^2 \begin{bmatrix} -\frac{1}{q_{max}^1} (1 - \frac{q^1}{q_{max}^1})^{-1} - \frac{1}{q_{min}^1} (\frac{q^1}{q_{min}^1} - 1)^{-1} \\ \vdots \\ -\frac{1}{q_{max}^i} (1 - \frac{q^i}{q_{max}^i})^{-1} - \frac{1}{q_{min}^i} (\frac{q^i}{q_{min}^i} - 1)^{-1} \\ \vdots \\ -\frac{1}{q_{max}^m} (1 - \frac{q^m}{q_{max}^m})^{-1} - \frac{1}{q_{min}^m} (\frac{q^m}{q_{min}^m} - 1)^{-1} \end{bmatrix} & \text{si } \mathbf{q} \notin \mathcal{L} \\ 0 & \text{si } \mathbf{q} \in \mathcal{L} \end{cases}$$

avec :

$$\begin{cases} \phi_{\mathbf{q}} = \log \left(\prod_{j=1}^m (1 - \frac{q^j}{q_{max}^j}) (\frac{q^j}{q_{min}^j} - 1) \right) \\ \nabla v_q^i = (2q^i - q_{max}^{li} - q_{min}^{li}) \prod_{\substack{j=1 \\ j \neq i}}^m (q_{max}^{lj} - q^j)(q^j - q_{min}^{lj}) \end{cases}$$

Dans le paragraphe suivant, nous définissons, de la même manière, un potentiel répulsif dans l'image afin de conserver les objets d'intérêt dans le champ de vision de la caméra.

6.2.6 Champ répulsif relatif à la contrainte de visibilité

Un point \mathcal{M}^j , qui se projette dans le plan image de la caméra en un point de coordonnée $\mathbf{p}^j = [u^j \ v^j \ 1]^T$, sera dit visible si $u_j \in [u_m \ u_M]$ et $v_j \in [v_m \ v_M]$, où u_m , u_M , v_m , v_M représentent les bords de l'image. Le vecteur \mathbf{s} contenant l'ensemble des coordonnées des points dans l'image sera dit acceptable si quel que soit $j \in \{1 \dots n\}$, $u^j \in [u_m + \alpha; u_M - \alpha]$ et $v^j \in [v_m + \alpha; v_M - \alpha]$, où α est une constante positive permettant de choisir la zone d'influence des limites de l'image (voir figure 6.3(a)). Ce paramètre doit être choisi tel que la force répulsive soit nulle pour la position désirée. Nous noterons \mathcal{C} l'ensemble regroupant les vecteurs \mathbf{s} acceptables. Afin de créer une barrière de potentiel autour du champ de vision de la caméra, assurant que tous les points considérés restent toujours visibles, et n'affectant pas les mouvements de la caméra quand tous ces points sont suffisamment éloignés des limites de l'image, on définit la fonction de potentiel $V_{\mathbf{s}}(\mathbf{s})$ directement dans l'espace image de la manière suivante (voir figure 6.3(b)) :

$$V_{\mathbf{s}}(\mathbf{s}) = \begin{cases} -v_s^2 \log \left(\prod_{j=1}^n \left(1 - \frac{u^j}{u_M}\right) \left(1 - \frac{u^j}{u_m}\right) \left(1 - \frac{v^j}{v_M}\right) \left(1 - \frac{v^j}{v_m}\right) \right) & \text{if } \mathbf{s} \notin \mathcal{C} \\ 0 & \text{if } \mathbf{s} \in \mathcal{C} \end{cases} \quad (6.27)$$

Comme pour la fonction de potentiel répulsif associée à l'évitement des butées articulaires, la fonction v_s est choisie bornée et nulle aux frontières de \mathcal{C} :

$$v_s(\mathbf{s}) = \prod_{j=1}^n (u^j - u_M^\alpha)(u^j - u_m^\alpha)(v^j - v_M^\alpha)(v^j - v_m^\alpha)$$

où $u_m^\alpha = u_m + \alpha$, $u_M^\alpha = u_M - \alpha$, $v_m^\alpha = v_m + \alpha$ et $v_M^\alpha = v_M - \alpha$.

La fonction $V_{\mathbf{s}}$ est positive ou nulle et tend vers l'infini quand au moins un des points considérés se rapproche des limites de l'image. Elle est nulle quand tous les points sont suffisamment éloignés des limites de l'image. La force artificielle répulsive dérivant de $V_{\mathbf{s}}$ est :

$$\mathbf{F}_{\mathbf{s}}(\mathbf{x}) = -\mathbf{M}^+ \mathbf{L}^+ \vec{\nabla}_{\mathbf{s}}^T V_{\mathbf{s}} \quad (6.28)$$

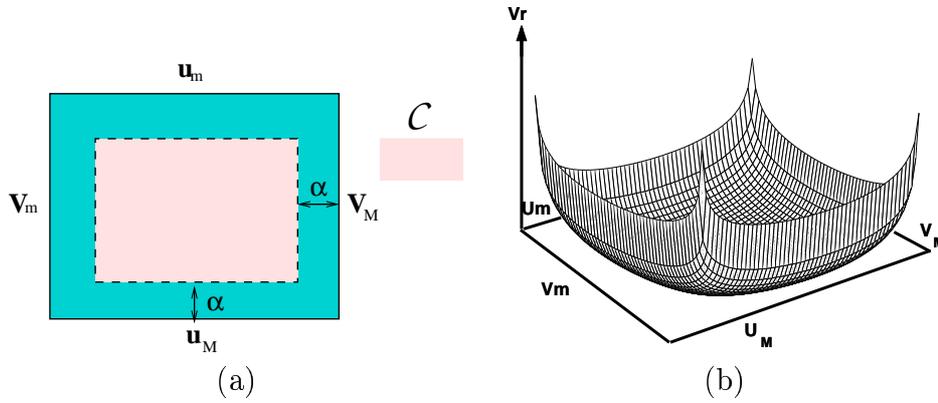


FIG. 6.3 – (a) limites de l'image, (b) fonction de potentiel associée à la contrainte de visibilité

où $\vec{\nabla}_s^T V_s$ est aisément obtenu à partir de la relation (6.27) :

$$\vec{\nabla}_s^T V_s = \begin{cases} 2 \begin{bmatrix} \nabla v_s^0 \\ \vdots \\ \nabla v_s^{2i} \\ \nabla v_s^{2i+1} \\ \vdots \\ \nabla v_s^{2n-1} \end{bmatrix} v_s \psi_s + v_s^2 \begin{bmatrix} -\frac{1}{u_M} \left(1 - \frac{u^1}{u_M}\right)^{-1} - \frac{1}{u_m} \left(1 - \frac{u^1}{u_m}\right)^{-1} \\ \vdots \\ -\frac{1}{u_M} \left(1 - \frac{u^i}{u_M}\right)^{-1} - \frac{1}{u_m} \left(1 - \frac{u^i}{u_m}\right)^{-1} \\ -\frac{1}{v_M} \left(1 - \frac{v^i}{v_M}\right)^{-1} - \frac{1}{v_m} \left(1 - \frac{v^i}{v_m}\right)^{-1} \\ \vdots \\ -\frac{1}{v_M} \left(1 - \frac{v^n}{v_M}\right)^{-1} - \frac{1}{v_m} \left(1 - \frac{v^n}{v_m}\right)^{-1} \end{bmatrix} & \text{si } s \notin \mathcal{C} \\ 0 & \text{si } s \in \mathcal{C} \end{cases}$$

avec :

$$\left\{ \begin{array}{l} \psi_s = \log \left(\prod_{j=1}^n \left(\left(1 - \frac{u^j}{u_M}\right) \left(1 - \frac{u^j}{u_m}\right) \left(1 - \frac{v^j}{v_M}\right) \left(1 - \frac{v^j}{v_m}\right) \right) \right) \\ \nabla v_s^{2i} = (2u^i - u_M^\alpha - u_m^\alpha)(v^i - v_M^\alpha)(v^i - v_m^\alpha) \prod_{\substack{j=1 \\ j \neq i}}^n (u^j - u_M^\alpha)(u^j - u_m^\alpha)(v^j - v_M^\alpha)(v^j - v_m^\alpha) \\ \nabla v_s^{2i+1} = (2v^i - v_M^\alpha - v_m^\alpha)(u^i - u_M^\alpha)(u^i - u_m^\alpha) \prod_{\substack{j=1 \\ j \neq i}}^n (u^j - u_M^\alpha)(u^j - u_m^\alpha)(v^j - v_M^\alpha)(v^j - v_m^\alpha) \end{array} \right.$$

Dans le paragraphe suivant, nous résumons le schéma de planification adoptée, dans les cas où les modèles des objets d'intérêt sont disponibles.

6.2.7 Résumé

En utilisant le modèle de l'objet d'intérêt et une caméra dont les paramètres de calibration sont connus, la trajectoire d'un ensemble de n points de l'image a été obtenue sous la forme d'une séquence de N vecteurs $\mathcal{S} = \{\mathbf{s}_k/k = 1 \cdots N\}$. Un schéma bloc du planificateur de trajectoire est donné par la figure 6.4. Rappelons qu'en pratique nous stoppons l'algorithme lorsque la distance entre les points courants dans l'image et les points désirés dans l'image est inférieur à une distance choisie par l'utilisateur. La trajectoire obtenue garantit les propriétés nécessaires à la réalisation de l'asservissement :

1. le long de la trajectoire planifiée, toutes les primitives visuelles considérées restent dans le champ de vision de la caméra;
2. le long de la trajectoire planifiée, aucun des axes du robot ne rencontre une de ses butées;
3. les mouvements du robot nécessaires au suivi des trajectoires dans l'image sont physiquement réalisables;
4. lorsque les potentiels répulsifs ne sont pas nécessaires, le centre optique de la caméra a un mouvement rectiligne.

En outre, l'ensemble $\mathcal{Z} = \{\mathbf{Z}_k = [Z_1 \cdots Z_n]/k = 1 \cdots N\}$, qui sera utilisé dans la loi de commande, a également été obtenu.

Dans la partie suivante, nous allons étendre cette méthode de planification de trajectoire au cas où le modèle de la cible n'est pas disponible. Dans ce cas de figure, on étudiera également l'influence des erreurs potentielles de calibration du capteur.

6.3 Cas d'un objet inconnu

Dans cette partie, on considère que le modèle de la cible n'est pas disponible. Dans ce cas, la pose de la caméra ne peut pas être estimée. On peut uniquement obtenir une reconstruction partielle du mouvement de la caméra (la rotation et la translation à un facteur d'échelle), en utilisant les méthodes issues de la géométrie épipolaire décrites dans le chapitre 2. L'estimation de la pose partielle et les formulations matricielles de la contrainte épipolaire sont exploitées pour construire la trajectoire des primitives visuelles dans l'image. Nous présentons, dans un premier temps, la méthode quand les paramètres de calibration sont précisément connus et ensuite nous étudions la robustesse de la méthode vis-à-vis des erreurs de modélisation. Afin de simplifier les notations, la matrice de rotation ${}^f\mathbf{R}_k$ et le vecteur de translation ${}^f\mathbf{t}_k$ seront notés \mathbf{R}_k et \mathbf{t}_k dans la suite.

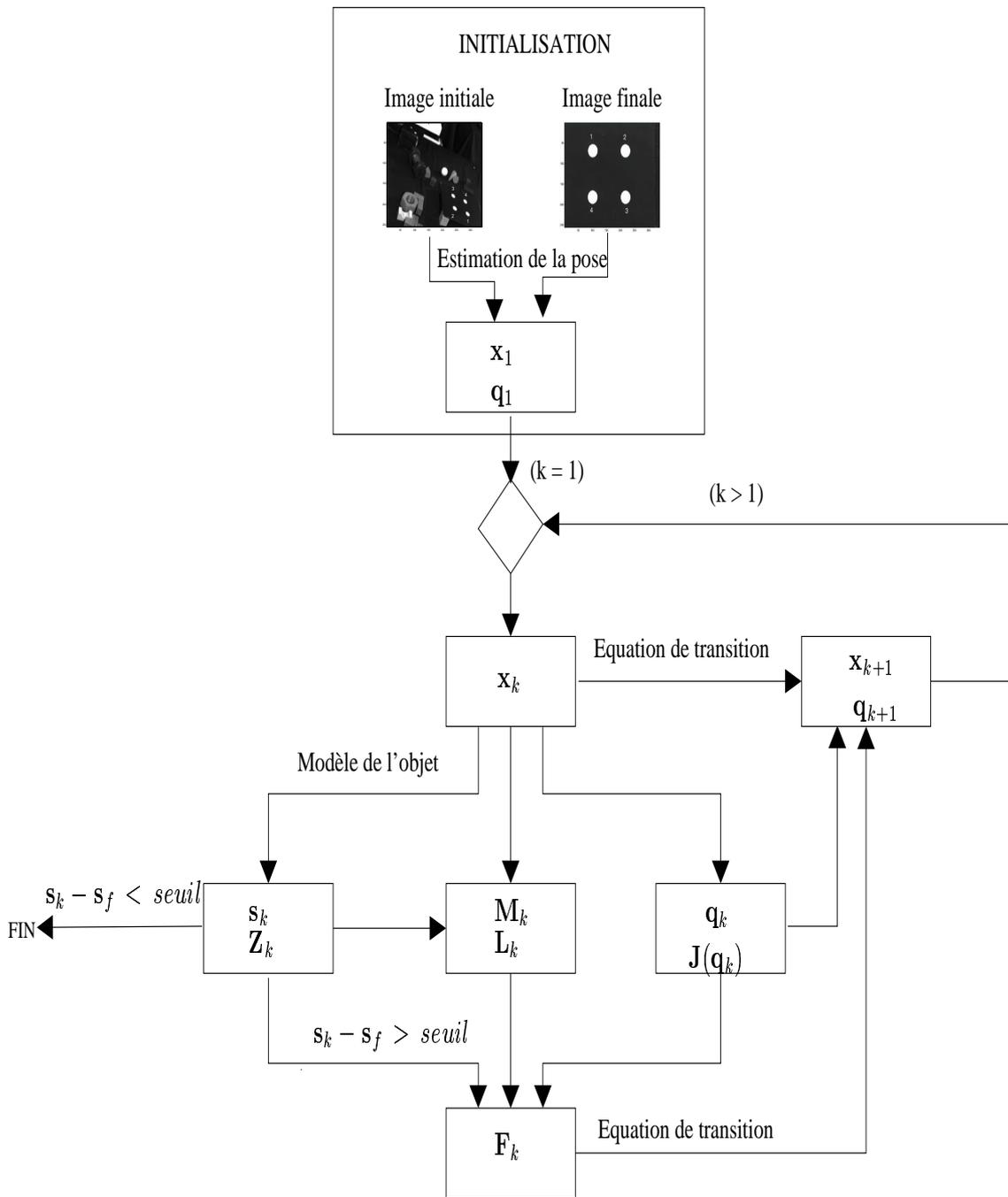


FIG. 6.4 – schéma bloc de la méthode de planification de trajectoire lorsque le modèle de la cible est disponible

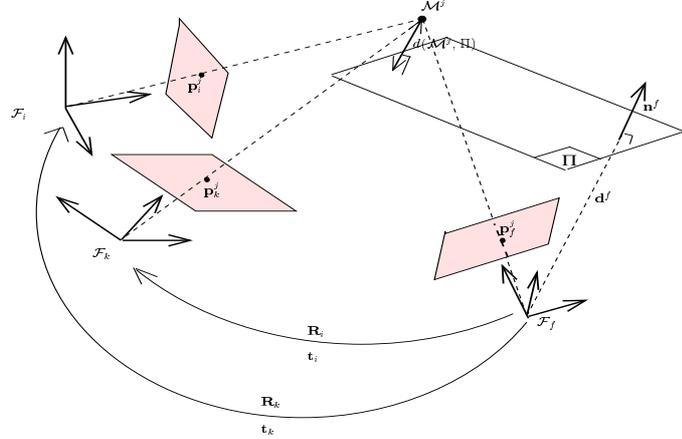


FIG. 6.5 – Trajectoire partielle de la caméra

6.3.1 Trajectoire partielle de la caméra

Nous choisissons ici le vecteur $\mathbf{x}_k = [\mathbf{t}_{dk}^T \boldsymbol{\theta}^T]^T$ pour représenter l'attitude de la caméra dans l'espace de travail \mathcal{W} à l'instant k , car dans ce cas la translation à un facteur d'échelle \mathbf{t}_{dk} est disponible alors que \mathbf{t}_k ne l'est pas. Nous avons donc $\mathbf{x}_i^T = [\mathbf{t}_{di}^T \boldsymbol{\theta}_i^T]$ quand la caméra est dans sa position initiale et $\mathbf{x}_f = \mathbf{0}_{1 \times 6}$ quand la caméra est dans la position désirée. A partir de la mise en correspondance des primitives visuelles dans l'image initiale et l'image désirée, il est possible de calculer la matrice de colinéation \mathbf{G}_i . Si les paramètres intrinsèques de la caméra sont connus, on peut remonter à la matrice d'homographie \mathbf{H}_i et ensuite estimer \mathbf{R}_i , \mathbf{t}_{di} , \mathbf{n}^f et donc \mathbf{x}_i . Comme dans le cas où le modèle de la cible était parfaitement connu, la trajectoire discrète \mathcal{X} débutant en \mathbf{x}_i est construite de manière itérative et est orientée dans la direction des forces artificielles données par :

$$\begin{cases} \mathbf{F}_x(\mathbf{x}) &= -\mathbf{x} \\ \mathbf{F}_s(\mathbf{x}) &= -\mathbf{M}^+ \mathbf{L}^+ \vec{\nabla}_s^T V_s \\ \mathbf{F}_q(\mathbf{x}) &= -\mathbf{M}^+ \mathbf{J} \vec{\nabla}_q^T V_q \end{cases}$$

Dans ce cas, le jacobien de paramétrisation est donné par :

$$\mathbf{M}(d^f) = \begin{bmatrix} d^f \mathbf{R}_k^T & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{L}_{wk}^+ \end{bmatrix} \quad (6.29)$$

La matrice \mathbf{M} dépend maintenant du paramètre inconnu d^f . Cependant, nous verrons dans la suite que ce paramètre n'a en fait aucune influence sur la planification de trajectoires si seule la contrainte de visibilité est considérée. En outre, la matrice d'interaction dépend de la profondeur Z_k^j de chaque point. Contrairement au cas précédent, où l'on disposait du modèle de la cible, ce paramètre ne peut pas directement être obtenu à partir de \mathbf{x}_k . Cependant, le ratio $\rho_k^j = Z_k^j/d^f$ peut aisément être estimé à partir de la paramétrisation choisie, en utilisant la relation (2.35). Nous écrirons donc une nouvelle fois la matrice d'interaction de la façon suivante :

$$\mathbf{L}(\mathbf{s}, \boldsymbol{\Sigma}, d^f) = \left[\frac{1}{d^f} \mathbf{S} \quad \mathbf{Q} \right] \quad (6.30)$$

où nous rappelons que $\boldsymbol{\Sigma}_k = [\rho_k^1 \cdots \rho_k^n]$ et où $\mathbf{S} = [\mathbf{S}^{1T} \cdots \mathbf{S}^{nT}]^T$ et $\mathbf{Q} = [\mathbf{Q}^{1T} \cdots \mathbf{Q}^{nT}]^T$ sont deux matrices de dimensions $2n \times 3$ indépendantes du paramètre d^f :

$$\left\{ \begin{array}{l} \mathbf{S}^j = \mathbf{a} \begin{bmatrix} -\frac{1}{\rho_k^j} & 0 & \frac{x_k^j}{\rho_k^j} \\ 0 & -\frac{1}{\rho_k^j} & \frac{y_k^j}{\rho_k^j} \end{bmatrix} \\ \mathbf{Q}^j = \mathbf{a} \begin{bmatrix} x_k^j y_k^j & -1 - x_k^{j2} & y_k^j \\ 1 + y_k^{j2} & -x_k^j y_k^j & -x_k^j \end{bmatrix} \end{array} \right.$$

La pose partielle de la caméra \mathbf{x}_k est donc obtenue à chaque itération. Ensuite, \mathbf{R}_k et \mathbf{t}_{dk} sont directement calculées à partir de \mathbf{x}_k . En utilisant les relations (2.35), le vecteur $\boldsymbol{\Sigma} = [\rho_k^1 \cdots \rho_k^n]$ est également obtenu à partir de \mathbf{x}_k . Les coordonnées articulaires du robot sont estimées de la même manière que dans le cas précédent :

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^+(\mathbf{q}_k) \mathbf{M}_k(d^f) (\mathbf{x}_{k+1} - \mathbf{x}_k)$$

Dans la section suivante, nous décrivons comment les trajectoires des primitives visuelles sont obtenues à partir des expressions matricielles de la contrainte épipolaire.

6.3.2 Trajectoire des primitives dans l'image

La matrice de colinéation \mathbf{G}_k relative au plan Π à l'instant k peut être calculée à partir de \mathbf{x}_k en utilisant les relations (2.17) et (2.31) :

$$\mathbf{G}_k = \mathbf{K}(\mathbf{R}_k^T - \mathbf{R}_k^T \mathbf{t}_{dk} \mathbf{n}^{fT}) \mathbf{K}^{-1} \quad (6.31)$$

Les coordonnées dans l'image de la projection des points \mathcal{M}^j à l'instant k sont alors obtenues en utilisant la relation (2.19) :

$$\alpha_k^j \mathbf{p}_k^j = [\alpha_k^j u_k^j \quad \alpha_k^j v_k^j \quad \alpha_k^j] = \mathbf{G}_k \mathbf{p}_f^j + \beta^j \mathbf{e}_k \quad (6.32)$$

où, d'après les relations (2.14), (2.23) et (2.21) :

$$\beta^j \mathbf{e}_k = \frac{d(\mathcal{M}^j, \Pi)}{Z_f^j d^f} \mathbf{K} \mathbf{R}_k^T \mathbf{t}_k = \frac{d(\mathcal{M}^j, \Pi)}{Z_f^j} \mathbf{K} \mathbf{R}_k^T \mathbf{t}_{dk} \quad (6.33)$$

En introduisant la relation (6.33) dans (6.32), on aboutit à :

$$\mu_k^j \mathbf{p}_k^j = \mathbf{G}_k \mathbf{p}_f^j + \frac{d(\mathcal{M}^j, \Pi)}{Z_f^j} \mathbf{K} \mathbf{R}_k^T \mathbf{t}_{dk} \quad (6.34)$$

En outre, en appliquant l'équation (6.34) entre les positions initiale et désirée de la caméra, on obtient aisément :

$$\frac{d(\mathcal{M}^j, \Pi)}{Z_f^j} = \text{sign} \left(\frac{(\mu_i^j \mathbf{p}_i^j - \mathbf{G}_i \mathbf{p}_f^j)_1}{(\mathbf{K} \mathbf{R}_i \mathbf{t}_{di})_1} \right) \frac{\|\mathbf{G}_i \mathbf{p}_f^j \wedge \mathbf{p}_i^j\|}{\|\mathbf{K} \mathbf{R}_i \mathbf{t}_{di} \wedge \mathbf{p}_i^j\|} \quad (6.35)$$

où si \mathbf{v} et b désignent respectivement un vecteur et un réel, alors $(\mathbf{v})_j$ représente la composante j de \mathbf{v} et $\text{sign}(b)$ représente le "signe" de b . Le paramètre $\frac{d(\mathcal{M}^j, \Pi)}{Z_f^j}$ est indépendant de la position courante de la caméra et pourra donc être calculé une fois pour toutes au début du processus de planification de trajectoire.

La relation (6.34) couplée aux relations (6.31) et (6.35) permet de calculer $\alpha_k^j \mathbf{p}_k^j$ à partir de \mathbf{x}_k et des vecteurs initial \mathbf{s}_i et désiré \mathbf{s}_f . Les coordonnées \mathbf{p}_k^j de la projection dans l'image de chaque point \mathcal{M}^j sont ensuite obtenues simplement en divisant $\alpha_k^j \mathbf{p}_k^j$ par sa dernière composante.

Dans la partie suivante, nous considérons que la force répulsive associée à l'évitement des butées articulaires du robot n'est pas activée (i.e $\chi_k = 0$). Nous étudions alors la répercussion, sur les trajectoires dans l'image, des éventuelles erreurs de calibration de la caméra et de l'utilisation d'une profondeur d^f erronée.

6.3.3 Influence du paramètre d^f

Si la force de répulsion associée à l'évitement des butées articulaires n'est pas activée, le paramètre d^f apparaît dans le processus de planification de trajectoire uniquement à travers la matrice \mathbf{W} définie comme le produit de $\mathbf{M}^+(d^f)$ et de $\mathbf{L}^+(\mathbf{s}, \Sigma, d^f)$. D'après les relations (6.29) et (6.30), cette matrice peut s'écrire :

$$\mathbf{W}(\mathbf{x}_k, \Sigma_k, \mathbf{s}_k) = (\mathbf{L}(\mathbf{s}_k, \Sigma_k, d^f) \mathbf{M}_k(d^f))^+ = \left[\begin{array}{c} \mathbf{S}(\mathbf{s}_k, \Sigma_k) \mathbf{R}_k^T \\ \mathbf{Q}(\mathbf{s}_k) \mathbf{L}_{wk}^+ \end{array} \right]^+$$

Donc \mathbf{W} est indépendante du paramètre d^f . Cela implique que les trajectoires des primitives visuelles dans l'image sont également indépendantes de la profondeur d^f . Notons que cela n'est malheureusement pas le cas lorsque la force artificielle associée à l'évitement des butées articulaires est également activée.

6.3.4 Influence de la calibration de la caméra

Si la caméra n'est pas parfaitement calibrée et que $\widehat{\mathbf{K}}$ est utilisée à la place de la matrice des paramètres intrinsèques réels, alors la matrice d'homographie estimée pour le plan de référence entre les images initiale et désirée, $\widehat{\mathbf{H}}_i$, s'écrit :

$$\widehat{\mathbf{H}}_i = \widehat{\mathbf{K}}^+ \mathbf{K} \mathbf{H}_i \mathbf{K}^+ \widehat{\mathbf{K}} = \delta \mathbf{K} \mathbf{H}_i \delta \mathbf{K}^+ \quad (6.36)$$

où $\delta \mathbf{K} = \widehat{\mathbf{K}}^+ \mathbf{K}$. Sous l'hypothèse que l'erreur initiale sur l'estimation de la matrice d'homographie se propage le long de la trajectoire :

$$\widehat{\mathbf{H}}_i = \delta \mathbf{K} \mathbf{H}_i \delta \mathbf{K}^+ \implies \widehat{\mathbf{H}}_k = \delta \mathbf{K} \mathbf{H}_k \delta \mathbf{K}^+ \quad (6.37)$$

la matrice d'homographie erronée $\widehat{\mathbf{H}}_k$, à l'instant k , peut être décomposée en la somme d'une matrice similaire à une matrice de rotation et d'une matrice de rang 1 :

$$\widehat{\mathbf{H}}_k = \widehat{\mathbf{H}}_{\infty k} - \widehat{\mathbf{T}}_k \widehat{\mathbf{n}}^{fT} \quad (6.38)$$

où $\widehat{\mathbf{H}}_{\infty k} = \delta \mathbf{K} \mathbf{R}_k^T \delta \mathbf{K}^+$, $\widehat{\mathbf{n}}^{fT} = \frac{\mathbf{n}^{fT} \delta \mathbf{K}}{\|\mathbf{n}^{fT} \delta \mathbf{K}^+\|}$ et $\widehat{\mathbf{T}}_k = \|\mathbf{n}^{fT} \delta \mathbf{K}^+\| \delta \mathbf{K} \mathbf{T}_k$ avec $\mathbf{T}_k = \mathbf{R}_k^T \mathbf{t}_{dk}$ [Malis 99]. En outre, \mathbf{G}_i , \mathbf{p}_f^j et \mathbf{p}_i^j ne sont pas affectés par d'éventuelles erreurs sur les paramètres intrinsèques, puisqu'ils sont directement obtenus à partir d'informations extraites de l'image exprimées en pixel.

D'après les relations (6.34), (6.35) et (6.38), les coordonnées homogènes $\widehat{\mathbf{p}}_k^j$ de la projection dans l'image des points \mathcal{M}^j , lorsque la calibration de la caméra est entachée d'erreurs, sont données par :

$$\widehat{\alpha}_k^j \widehat{\mathbf{p}}_k^j = \widehat{\mathbf{K}} \widehat{\mathbf{H}}_k \widehat{\mathbf{K}}^+ \mathbf{p}_f^j + \text{sign} \left(\frac{(\alpha_i^j \mathbf{p}_i^j - \mathbf{G}_i \mathbf{p}_f^j)_1}{(\widehat{\mathbf{K}} \widehat{\mathbf{T}}_i)_1} \right) \frac{\|\mathbf{G}_i \mathbf{p}_f^j \wedge \mathbf{p}_i^j\|}{\|\widehat{\mathbf{K}} \widehat{\mathbf{T}}_i \wedge \mathbf{p}_i^j\|} \widehat{\mathbf{K}} \widehat{\mathbf{T}}_k \quad (6.39)$$

De plus, en remarquant que :

$$\widehat{\mathbf{K}} \widehat{\mathbf{T}}_k = \|\mathbf{n}^{fT} \delta \mathbf{K}^+\| \mathbf{K} \mathbf{T}_k$$

nous obtenons les relations :

$$\left\{ \begin{array}{l} \text{sign} \left(\frac{(\alpha_i^j \mathbf{p}_i^j - \mathbf{G}_i \mathbf{p}_f^j)_1}{(\widehat{\mathbf{K}} \widehat{\mathbf{T}}_i)_1} \right) = \text{sign} \left(\frac{(\alpha_i^j \mathbf{p}_i^j - \mathbf{G}_i \mathbf{p}_f^j)_1}{(\mathbf{K} \mathbf{T}_i)_1} \right) \\ \frac{\|\mathbf{G}_i \mathbf{p}_f^j \wedge \mathbf{p}_i^j\|}{\|\widehat{\mathbf{K}} \widehat{\mathbf{T}}_i \wedge \mathbf{p}_i^j\|} \widehat{\mathbf{K}} \widehat{\mathbf{T}}_k = \frac{\|\mathbf{G}_i \mathbf{p}_f^j \wedge \mathbf{p}_i^j\|}{\|\mathbf{K} \mathbf{T}_i \wedge \mathbf{p}_i^j\|} \mathbf{K} \mathbf{T}_k \end{array} \right. \quad (6.40)$$

Par ailleurs, nous avons également la propriété suivante :

$$\widehat{\mathbf{K}} \widehat{\mathbf{H}}_k \widehat{\mathbf{K}}^+ = \widehat{\mathbf{K}} \delta \mathbf{K} \mathbf{H}_k \delta \mathbf{K}^+ \widehat{\mathbf{K}}^+ = \mathbf{K} \mathbf{H}_k \mathbf{K}^+ \quad (6.41)$$

Finalement, en introduisant les relations (6.40) et (6.41) dans (6.39), nous obtenons :

$$\widehat{\alpha}_k^j \widehat{\mathbf{p}}_k^j = \alpha_k^j \mathbf{p}_k^j$$

Sous l'hypothèse (6.37), les trajectoires dans l'image des primitives visuelles ne sont donc pas affectées par une mauvaise calibration de la caméra. Rappelons que la relation (6.37) signifie que l'erreur initiale se propage le long de la trajectoire planifiée. Cette relation est exacte si la matrice de colinéation à l'infini à l'instant initial est connue. Nous vérifierons cette propriété très intéressante de manière expérimentale sur la plate-forme robotisée de l'IRISA.

6.3.5 Résumé

Dans cette partie, les trajectoires d'un ensemble de n points dans l'image ont été obtenues sous la forme d'une séquence ordonnée de N vecteurs $\mathcal{S} = \{\mathbf{s}_k/k = 1 \cdots N\}$. La méthode proposée ne nécessite pas le modèle géométrique des objets 3D. L'ensemble $\mathcal{R} = \{\boldsymbol{\Sigma}_k/k = 1 \cdots N\}$, qui sera utilisé dans la phase d'asservissement, a également été obtenu.

Comme dans le cas où le modèle des objets d'intérêt est connu, le long de la trajectoire planifiée toutes les primitives visuelles considérées restent dans le champ de vision de la caméra, le robot n'atteint pas ces limites mécaniques, les mouvements nécessaires au suivi des trajectoires dans l'image sont physiquement réalisables et le centre optique de la caméra a un mouvement rectiligne hors des zones de répulsion. En outre, nous avons vu que la méthode proposée est robuste vis-à-vis des erreurs sur la calibration de la caméra et sur la profondeur d^f quand la force artificielle associée à l'évitement des butées articulaires n'est pas nécessaire.

Un schéma bloc du planificateur de trajectoire est donné par la figure 6.6.

6.4 Interpolation par des fonctions B-splines

Dans les parties précédentes, nous avons obtenu des trajectoires discrètes des primitives dans l'image. Afin de construire des trajectoires continues et dérivables, et ainsi améliorer le comportement de la commande, nous utilisons une interpolation par B-spline cubique des points dans l'image. En utilisant un tel procédé, la trajectoire interpolée entre deux points successifs peut en théorie conduire à des positions de la caméra physiquement inatteignable (voire même inexistantes). Il se peut donc que les mouvements de la caméra soient impossibles à réaliser pour atteindre une position interpolée dans l'image, ce qui entraîne le système vers un minimum local avec un asservissement visuel classique où la consigne est fixe [Chaumette 98]. Cependant, la distribution des points dans l'image peut être choisie suffisamment dense pour que ces problèmes potentiels n'apparaissent pas. En effet la distribution est choisie par l'utilisateur en fixant le paramètre ε_k dans la relation (6.4). En pratique,

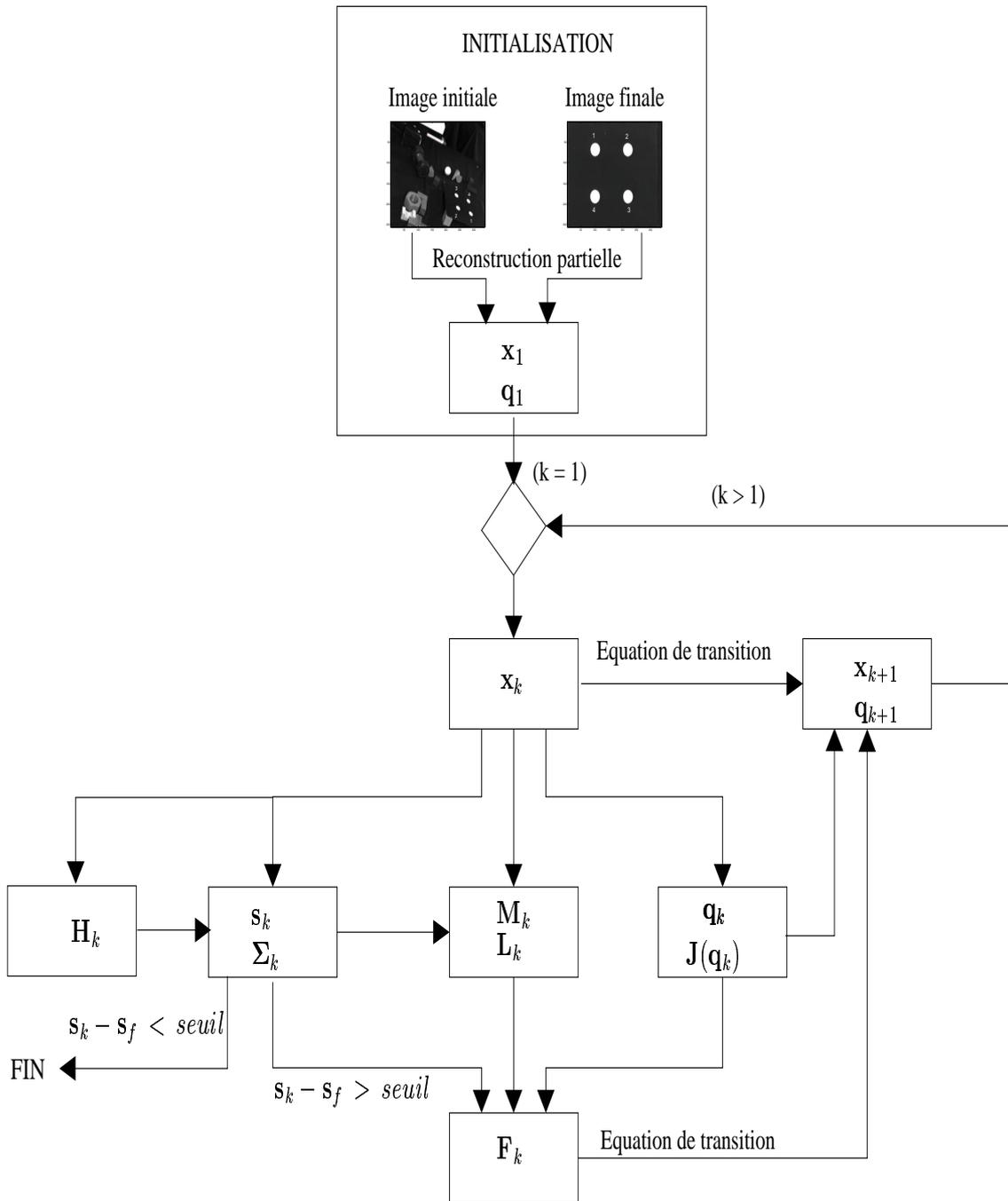


FIG. 6.6 – Schéma bloc de la méthode de planification de trajectoire lorsque le modèle de la cible n'est pas disponible

nous avons utilisé de 350 à 800 points lors de nos expérimentations. L'ensemble de ces points correspond à des positions physiquement valides où le jacobien de la tâche est régulier. Entre deux points succesifs suffisamment proche, l'existence potentielle de mouvements impossibles à réaliser ne peut se produire que pendant un moment très court, et la variation, à chaque instant, de la consigne permet d'éviter de rester dans un minimum local. Si la tâche reste admissible le long de la trajectoire interpolée alors il n'est pas nécessaire que ces trajectoires correspondent exactement à des points de l'objet observé car la régulation de fonction de tâche est dans ce cas assurée.

Le problème d'interpolation par des fonctions B-splines est en général posé de la manière suivante [Schumaker 81] :

Etant donné un ensemble de points $\mathcal{S} = \{\mathbf{s}_k/k = 1 \cdots N\}$ et un ensemble de paramètres $\mathcal{T} = \{t_k/k = 1 \cdots N\}$, nous désirons construire une fonction B-spline cubique $\mathbf{s}(t)$ telle que $\mathbf{s}(t_k) = \mathbf{s}_k$.

En pratique, les valeurs des paramètres sont rarement disponibles. Dans notre cas, nous pouvons les ajuster en utilisant la distribution des vecteurs de primitives images \mathbf{s}_k ou en utilisant la distribution des attitudes de la caméra \mathbf{x}_k . La distance entre deux attitudes successives de la caméra \mathbf{x}_k et \mathbf{x}_{k+1} peut être choisie constante en fixant le paramètre ε_k à une valeur constante. Notons qu'en procédant de cette manière, la distance dans l'image entre un point à l'instant k et un point à l'instant $k+1$ n'est pas nécessairement constante (voir figure 6.7). Afin de contrôler efficacement la vitesse de la caméra, les paramètres temporels t_k et t_{k+1} sont choisis espacés proportionnellement à la distance entre les attitudes \mathbf{x}_k et \mathbf{x}_{k+1} de la caméra. Le temps entre deux itérations est donc également constant : $\Delta t_{k+1} = t_{k+1} - t_k = T$, où T peut par exemple être choisi égal à la période d'échantillonnage de la caméra. Le problème posé est alors : étant donné :

- les N vecteurs \mathbf{s}_k de dimensions $2n$
- les N valeurs des paramètres temporels t_k

trouver les matrices diagonales de dimension $2n \times 2n$, \mathbf{A}_k , \mathbf{B}_k , \mathbf{C}_k et \mathbf{D}_k , telles que :

$$\mathbf{s}(t) = \mathbf{A}_k t^3 + \mathbf{B}_k t^2 + \mathbf{C}_k t + \mathbf{D}_k \quad \text{pour } (k-1)T \leq t < kT \quad (6.42)$$

Nous obtenons une fonction, $\mathbf{s}(t)$, C^2 avec $\mathbf{s}(k\Delta T) = \mathbf{s}_k$ pour $k = 1 \cdots N$, en utilisant un processus d'interpolation par B-splines naturelles. Les matrices \mathbf{A}_k , \mathbf{B}_k , \mathbf{C}_k et \mathbf{D}_k sont alors directement obtenues à partir de \mathcal{S} et \mathcal{T} [Schumaker 81]. La profondeur Z (respectivement le ratio ρ) dans le cas où l'objet d'intérêt est connu (respectivement dans le cas où l'objet d'intérêt est inconnu) apparaît dans la loi de commande. En utilisant le même procédé que pour l'obtention de $\mathbf{s}(t)$, les fonctions $\mathbf{Z}(t) = [Z^1(t) \cdots Z^n(t)]$ (respectivement $\mathbf{\Sigma}(t) = [\rho^1(t) \cdots \rho^n(t)]$) sont obtenues à partir de $\mathcal{Z} = \{\mathbf{Z}_k/k = 1 \cdots N\}$ et \mathcal{T} (respectivement $\mathcal{R} = \{\mathbf{\Sigma}_k/k = 1 \cdots N\}$ et \mathcal{T}).

Dans la section suivante, nous utilisons un asservissement visuel 2D afin de suivre les trajectoires obtenues dans l'espace de tâche.

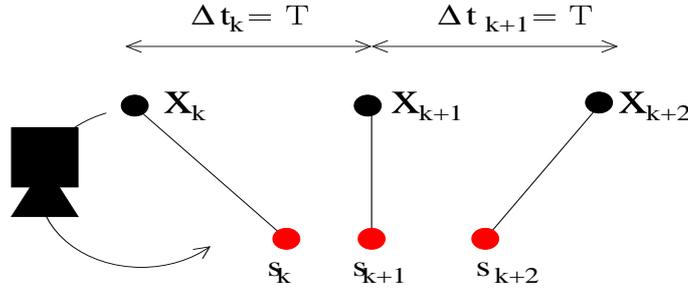


FIG. 6.7 – Interpolation par B-spline cubique

6.5 Suivi des trajectoires par asservissement visuel 2D

Pour effectuer le suivi de trajectoire, nous utilisons un asservissement visuel 2D. Comme dans le chapitre précédent, la loi de commande obtenue est composée d'un terme de compensation proportionnelle de l'erreur entre la mesure courante et sa valeur désirée sur la trajectoire planifiée et d'un terme de compensation de l'erreur de traînage. Ces deux termes pourront être estimés en utilisant la forme continue et dérivable des trajectoires obtenue après interpolation.

Nous définissons, une nouvelle fois, la fonction de tâche \mathbf{e} à réguler à $\mathbf{0}$ de la manière suivante :

$$\mathbf{e} = \hat{\mathbf{L}}^+(\mathbf{s}(\mathbf{r}(t)) - \mathbf{s}^*(t))$$

Le vecteur $\mathbf{s}^*(t)$ est la trajectoire désirée des primitives visuelles \mathbf{s} , obtenue en utilisant les méthodes décrites précédemment. La matrice $\hat{\mathbf{L}}^+$ est la pseudo-inverse d'un modèle choisi de la matrice d'interaction \mathbf{L} associée à \mathbf{s} . La valeur de \mathbf{L} à la position courante désirée est utilisée pour $\hat{\mathbf{L}}$. Plus précisément :

- si la cible est connue $\hat{\mathbf{L}} = \mathbf{L}(\mathbf{s}^*(t), \mathbf{Z}^*(t))$
- sinon $\hat{\mathbf{L}} = \mathbf{L}(\mathbf{s}^*(t), \mathbf{\Sigma}^*(t), \hat{d}^f)$, \hat{d}^f étant une valeur estimée de d^f (voir la relation (6.30)) :

$$\mathbf{L}(\mathbf{s}^*(t), \mathbf{\Sigma}^*(t), \hat{d}^f) = \begin{bmatrix} \frac{1}{\hat{d}^f} \mathbf{S}(\mathbf{s}^*(t), \mathbf{\Sigma}^*(t)) & \mathbf{Q}(\mathbf{s}^*(t)) \end{bmatrix}$$

Comme nous l'avons déjà vu, si on désire une décroissance exponentielle de la fonction de tâche \mathbf{e} vers $\mathbf{0}$ et si les objets d'intérêt sont immobiles, alors nous avons

$\frac{\partial \mathbf{e}}{\partial t} = -\widehat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t}$ et la commande en vitesse est donnée par la relation :

$$\boldsymbol{\tau}_c = -\lambda \mathbf{e} + \widehat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t}$$

où le terme $\widehat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t}$ permet de compenser les erreurs de suivi. Plus précisément, nous avons à partir de (6.42) :

$$\frac{\partial \mathbf{s}^*}{\partial t} = 3\mathbf{A}_k t^2 + 2\mathbf{B}_k t + \mathbf{C}_k \quad \text{pour } (k-1)T \leq t < kT$$

Comme nous allons le voir dans la partie consacrée aux résultats expérimentaux, cette commande en vitesse est très robuste vis-à-vis des erreurs sur les paramètres de calibration et sur la profondeur d^f . En effet, l'erreur entre les valeurs courante et désirée des primitives images, utilisée en entrée du processus de régulation, reste suffisamment faible pour assurer un très bon comportement du système.

6.6 Résultats expérimentaux

Dans cette partie, nous présentons les résultats expérimentaux relatifs aux méthodes de planification de trajectoire dans l'image dans le cas d'un objet connu ou non. Nous présenterons les résultats obtenus avec une caméra calibrée et ceux obtenus avec des paramètres erronés.

6.6.1 Conditions expérimentales

Les méthodes proposées ont été testées sur la plate-forme robotique de l'IRISA, constituée d'un robot cartésien AFMA à six degrés de liberté. La caméra est embarquée sur l'organe terminal du robot et observe un objet d'intérêt statique. La tâche consiste à repositionner le robot de telle sorte que l'image acquise par la caméra à la fin de l'asservissement corresponde à une image acquise lors d'une phase d'apprentissage de l'image désirée. Nous avons utilisé deux types de cible :

- un objet plan composé de quatre disques blancs sur fond noir;
- un objet non plan composé de neuf disques blancs sur fond noir.

Les informations visuelles utilisées sont les coordonnées du centre de gravité de chaque disque. Par ailleurs, l'ensemble du dispositif a été calibré (jacobien du robot, matrice de passage caméra-effecteur, paramètres intrinsèques de la caméra).

Dans les sections 6.6.2, 6.6.3 et 6.6.4, les tâches à réaliser ne nécessitent pas l'introduction des contraintes liées à l'évitement des butées articulaires du robot. Dans la section 6.6.5, nous commentons les résultats obtenus quand la tâche à réaliser nécessite à la fois l'intégration de la contrainte de visibilité et celle liée à l'évitement des butées articulaires.

6.6.2 Cas d'une cible plane connue

Dans cette partie, nous présentons les résultats obtenus avec une cible plane et en supposant que son modèle est parfaitement connu. Les images initiale et désirée de la scène sont illustrées sur les figures 6.8(a) et 6.8(b) respectivement. Le déplacement entre ces deux prises de vue est très important et correspond à :

translation (mm)	$t_x = 300$	$t_y = 55$	$t_z = 120$
rotation (dg)	$(u\theta)_x = 28$	$(u\theta)_y = 78$	$(u\theta)_z = 147$

Dans ce cas de figure, les approches classiques $3D$ et $2D$ ne permettent pas la réalisation de la tâche de positionnement. Pour mettre en évidence l'importance de la contrainte de visibilité, nous avons réalisé la phase de planification de trajectoires en désactivant le potentiel répulsif qui est associé à cette contrainte. Les trajectoires obtenues, qui correspondent aux trajectoires dans l'image qu'aurait généré un asservissement visuel $3D$, sont fournies dans la figure 6.9. Nous observons que les primitives visuelles quittent le champ de vision de la caméra. Naturellement, de telles trajectoires ne sont pas souhaitables puisqu'elles ne permettent pas la réalisation de la tâche.

Les résultats de la figure 6.10 ont été obtenus en utilisant les paramètres de la caméra fournis par le constructeur. La force répulsive liée à la contrainte de visibilité est alors activée. Les primitives $3D$ utilisées pour l'initialisation du processus de planification de trajectoire ont été obtenues en couplant les algorithmes de Dementhon [Dementhon 95] et de Lowe [Lowe 91]. Les trajectoires planifiées en utilisant l'algorithme illustré sur la figure 6.4 sont données par la figure 6.10(a). On remarque que cette fois toutes les primitives visuelles considérées restent dans le champ de vision de la caméra. La tâche peut alors être réalisée. Les trajectoires réalisées sont fournies sur la figure 6.10(b). Il faut souligner que ces trajectoires sont également satisfaisantes et similaires aux trajectoires planifiées. Cela montre l'efficacité du schéma de commande. L'erreur de traînage ($\mathbf{s}(t) - \mathbf{s}^*(t)$) est représentée par la figure 6.10(d). On remarque que cette erreur reste suffisamment faible (l'erreur maximale reste inférieure à 5 pixels) pour assurer le bon comportement de la loi de commande $2D$. Les erreurs sur les coordonnées des points dans l'image sont représentées sur la figure 6.10(c). La convergence des coordonnées dans l'image de chacun des points considérés vers leur valeur désirée, démontre que la tâche a été correctement réalisée. Les vitesses de rotation et de translation de la caméra sont fournies par la figure 6.10(e). On remarque que les variations de la vitesse de la caméra sont très satisfai-

santes en raison de la régularité de la fonction de tâche. Notons finalement que la trajectoire de la caméra, donnée par la figure 6.10(f), est tout à fait correcte.

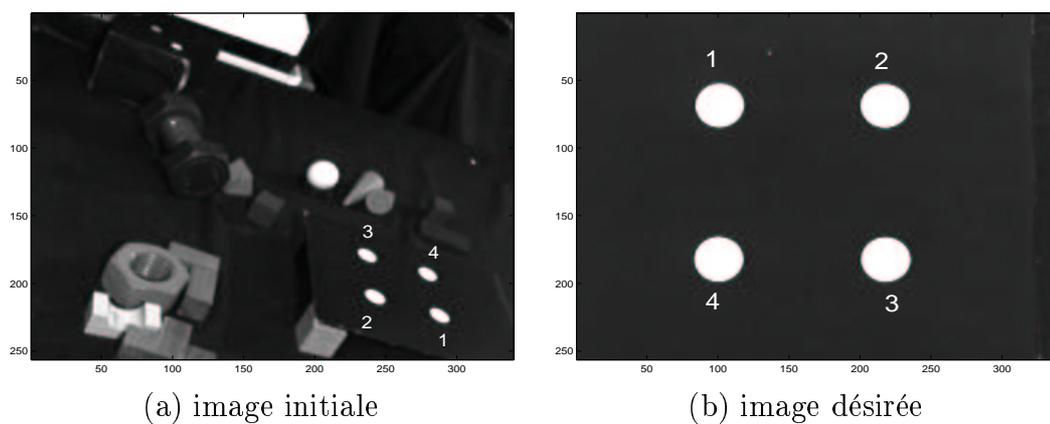


FIG. 6.8 – Image de la cible plane pour la première expérience

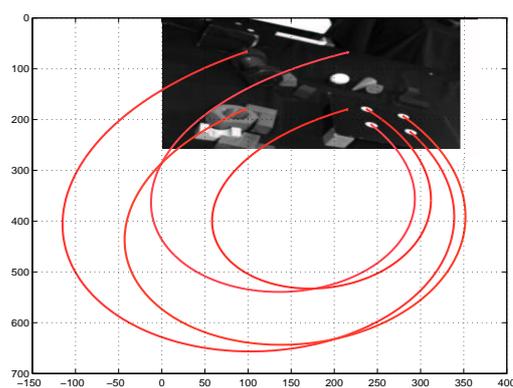
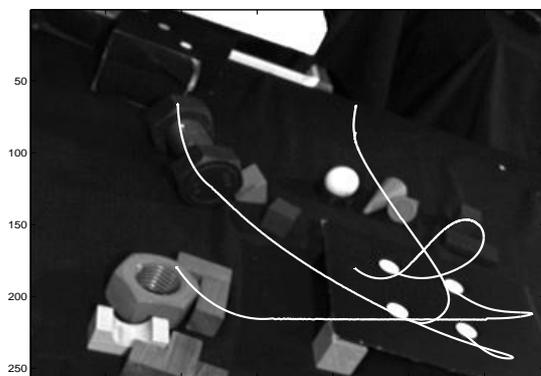
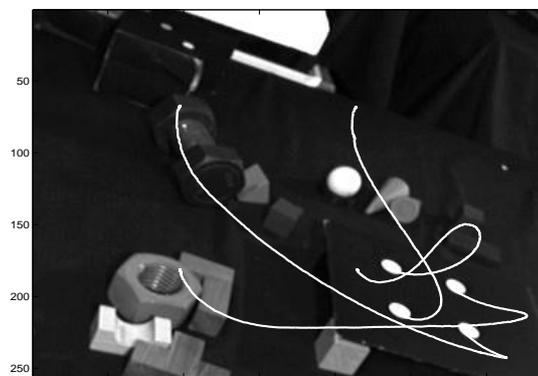


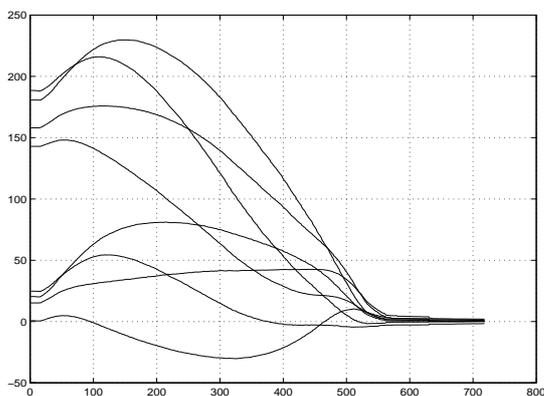
FIG. 6.9 – Trajectoires dans l'image quand la force répulsive n'est pas activée



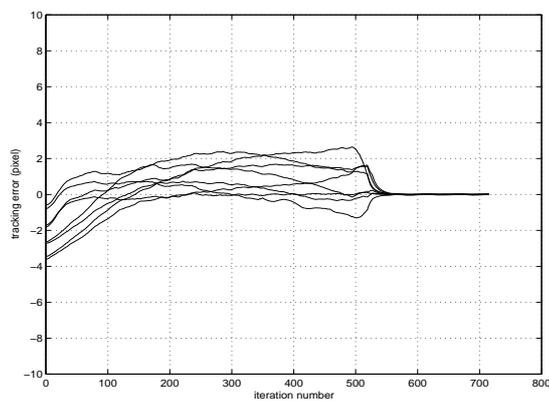
(a) trajectoires planifiées



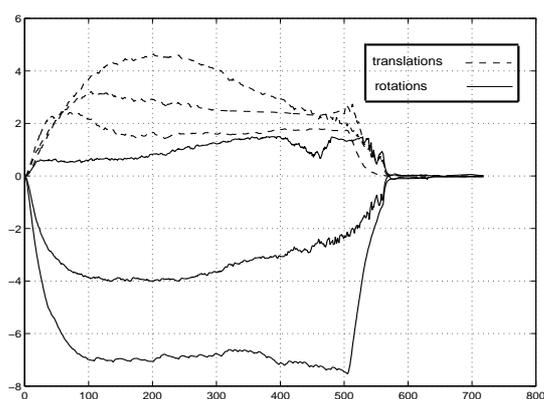
(b) trajectoires suivies



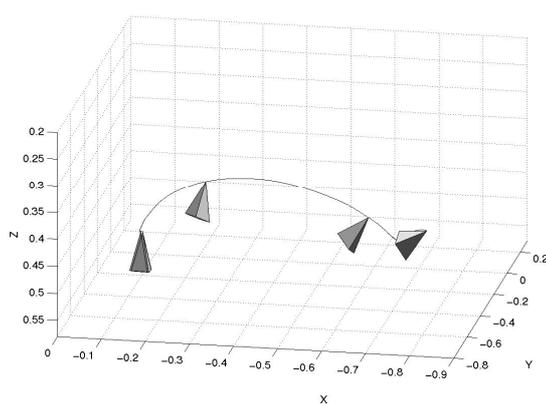
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel



(e) vitesse de la caméra (cm/s et deg/s)



(f) trajectoire de la caméra

FIG. 6.10 – Résultat dans le cas où le modèle de l'objet est connu

6.6.3 Cas d'une cible plane inconnue

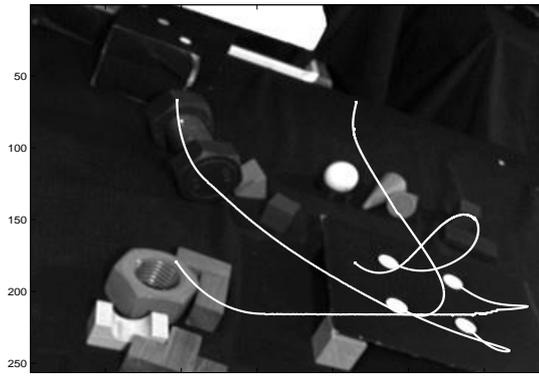
Nous présentons à présent les résultats expérimentaux obtenus en considérant que le modèle de l'objet est inconnu. Les images initiale et désirée utilisées sont les mêmes que pour l'expérimentation précédente. Comme nous l'avons déjà remarqué, dans le cas où l'on ne dispose pas du modèle de la cible, seule une estimation de la pose partielle de la caméra (i.e. la rotation et la translation à un facteur d'échelle près) peut être obtenue. La pose partielle de la caméra entre son attitude initiale et son attitude désirée, utilisée pour initialiser l'algorithme de planification de trajectoires (voir le schéma bloc 6.6), est obtenue à partir de l'estimation de la géométrie épipolaire entre les deux images considérées. Pour vérifier la robustesse de notre méthode de planification de trajectoire, nous présentons les résultats obtenus en utilisant trois ensembles de paramètres :

1. calibration correcte : les paramètres intrinsèques sont correctement estimés et la profondeur d^f est prise égale à sa valeur réelle de 35 cm, les résultats de cette expérience sont reportés sur la figure 6.11;
2. mauvaise calibration : dans ce cas, une erreur de 20% a été ajoutée à chacun des paramètres intrinsèques et la profondeur d^f a été prise égale à 20 cm, les résultats de cette expérience sont reportés sur la figure 6.12;
3. très mauvaise calibration : dans ce dernier cas, une erreur de 50% a été ajoutée à chacun des paramètres intrinsèques et la profondeur d^f a été prise égale à 70 cm, les résultats de cette expérience sont reportés sur la figure 6.13;

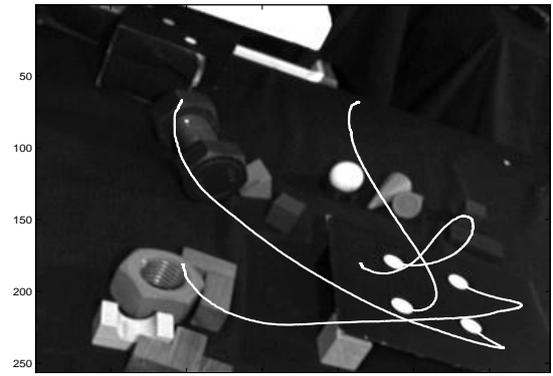
1) **Calibration correcte** : Soulignons, tout d'abord, que les trajectoires obtenues avec ou sans modèle de la cible sont très similaires (voir les figures 6.10 et 6.11). Une nouvelle fois, les trajectoires planifiées représentées sur la figure 6.11(a) sont satisfaisantes et permettent la réalisation de la tâche de positionnement. Les trajectoires effectivement réalisées, données par la figure 6.11(b), sont similaires aux trajectoires planifiées. Les erreurs de traînage sur les coordonnées dans l'image de chaque point sont fournies par la figure 6.11(d). Ces erreurs restent faibles durant toute la phase d'asservissement (toujours inférieures à 5 pixels). Cela confirme, de nouveau, le bon comportement de la commande. La tâche est correctement réalisée, ce qui peut être vérifié en notant la convergence vers 0 de l'erreur des coordonnées dans l'image de tous les points considérés (voir figure 6.11(e)).

2) et 3) **Mauvaise et très mauvaise calibrations** : Nous testons ici la robustesse de notre méthode vis-à-vis des erreurs sur l'estimation de la calibration et de la profondeur d^f . Comme nous pouvons le constater sur les figures 6.11(a) et 6.12(a), les trajectoires planifiées dans les cas de la calibration correcte et de la mauvaise calibration sont similaires. Cela confirme la robustesse de l'algorithme de planification de trajectoire vis-à-vis des erreurs considérées. Notons également que les trajectoires planifiées dans le cas de la très mauvaise calibration, représentées

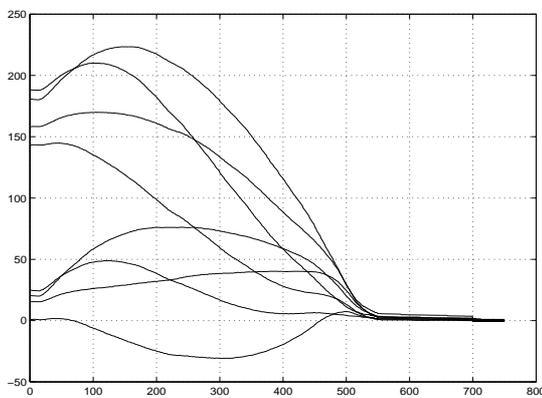
sur la figure 6.13(a), sont également proches de celles obtenues dans les autres cas. Les trajectoires réalisées (voir figures 6.12(b) et 6.13(b)) sont elles-aussi très satisfaisantes. Les erreurs de traînage fournies par la figure 6.12(d), dans le cas de la mauvaise calibration, restent faibles durant l'asservissement (toujours inférieur à 5 pixels). Dans le cas d'une très mauvaise calibration, celles-ci sont représentées par la figure 6.13(d) et restent toujours inférieures à 10 pixels. Dans ces configurations, l'asservissement visuel $2D$ fournit de très bons résultats et fait preuve d'une grande robustesse. Dans les deux expérimentations, la tâche de positionnement est correctement réalisée comme le démontre la convergence vers 0 de l'erreur des coordonnées dans l'image de tous les points considérés (voir figures 6.12(e) et 6.13(e)).



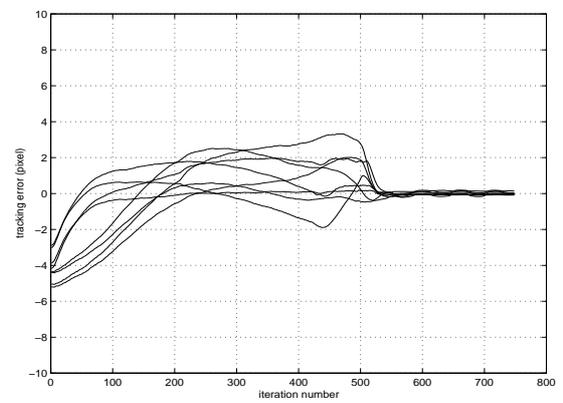
(a) trajectoires planifiées



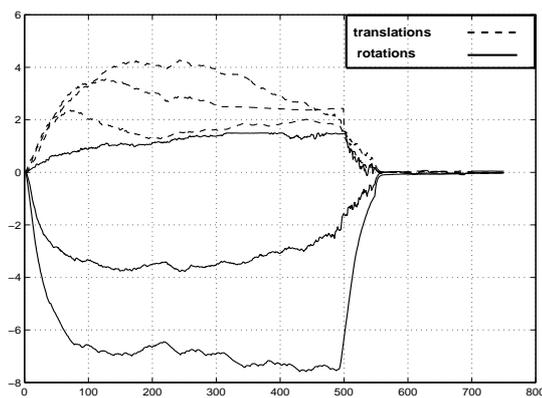
(b) trajectoires suivies



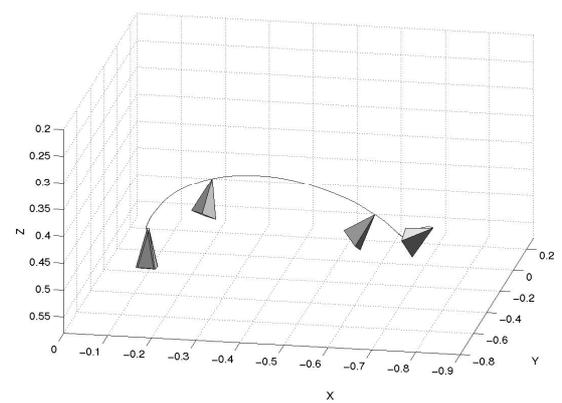
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

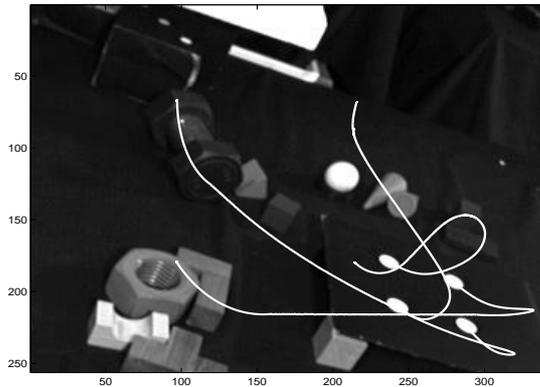


(e) vitesse de la caméra (cm/s et deg/s)

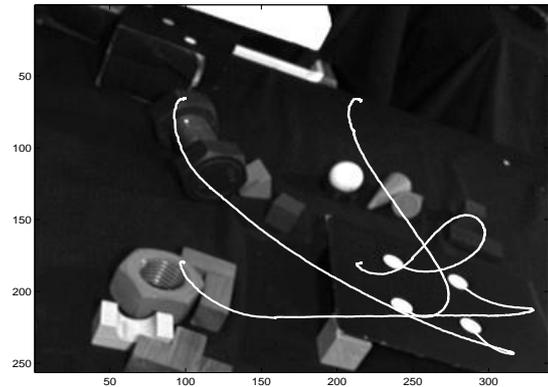


(f) trajectoire de la caméra

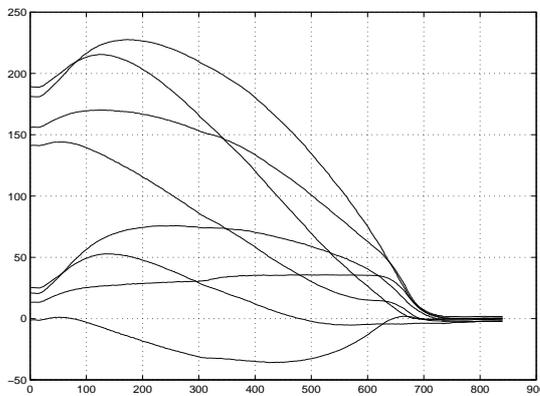
FIG. 6.11 – Résultat dans le cas où le modèle de l'objet est inconnu et pour une calibration correcte



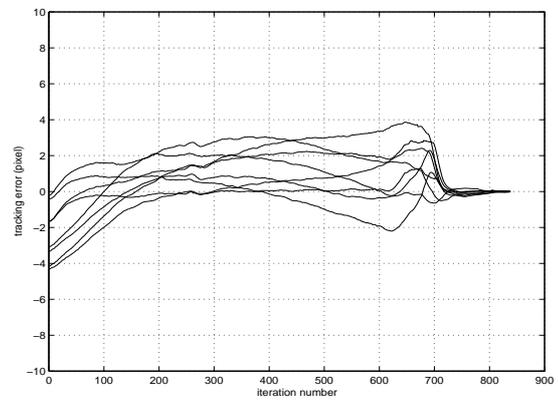
(a) trajectoires planifiées



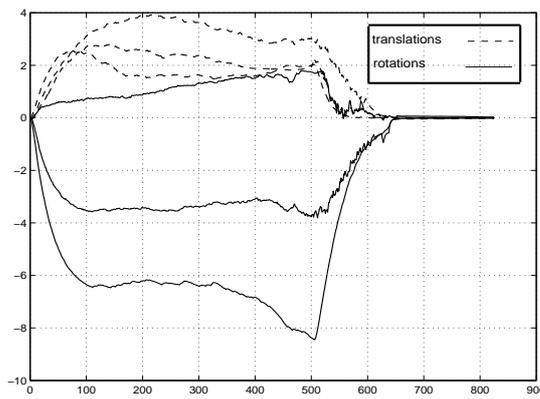
(b) trajectoires suivies



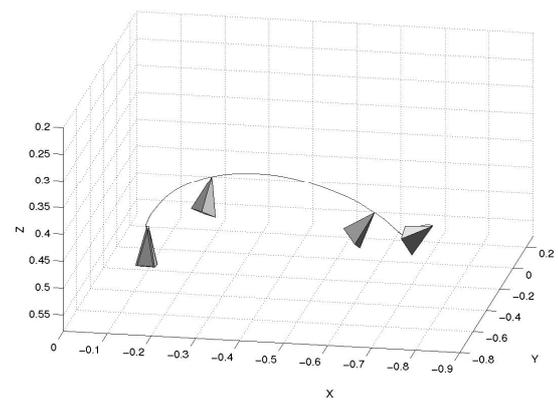
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel

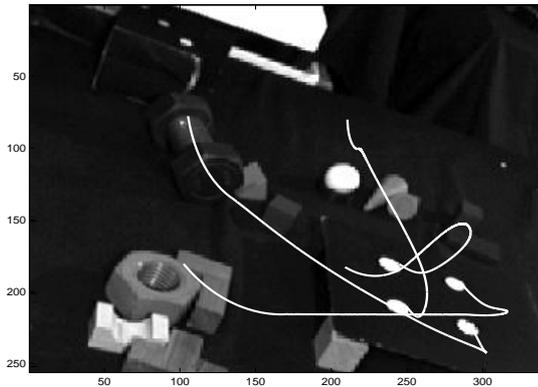


(e) vitesse de la caméra (cm/s et deg/s)

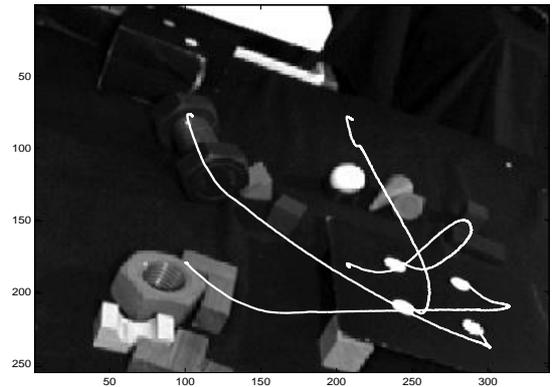


(f) trajectoire de la caméra

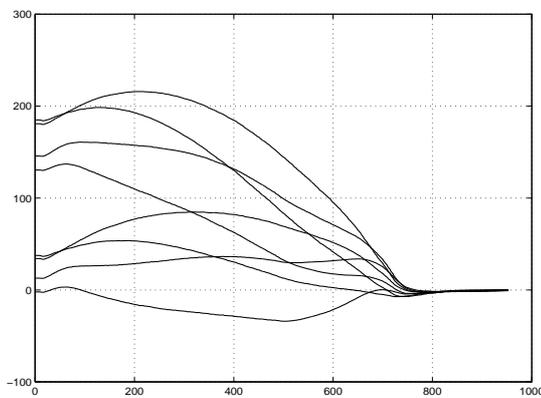
FIG. 6.12 – Résultat dans le cas où le modèle de l'objet est inconnu et pour une mauvaise calibration



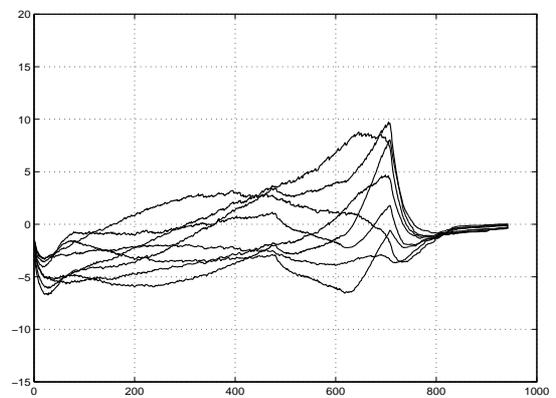
(a) trajectoires planifiées



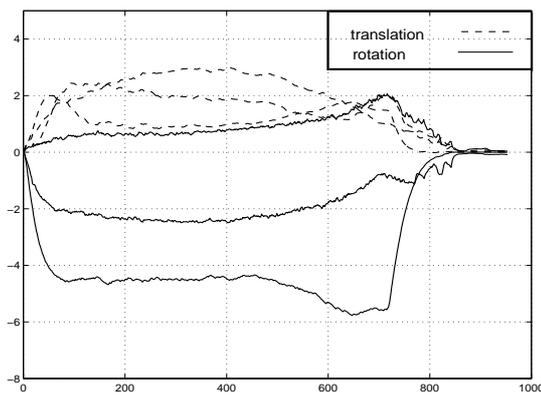
(b) trajectoires suivies



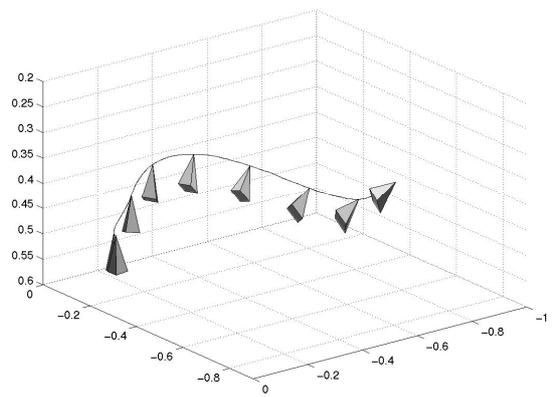
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel



(e) vitesse de la caméra (cm/s et deg/s)



(f) trajectoire de la caméra

FIG. 6.13 – Résultat dans le cas où le modèle de l'objet est inconnu et pour une très mauvaise calibration

6.6.4 Résultats pour un objet non-planaire et inconnu

La cible est maintenant non planaire. Une nouvelle fois le déplacement à effectuer est très important :

translation (mm)	$t_x = -672$	$t_y = -1062$	$t_z = 468$
rotation (dg)	$(u\theta)_x = 30.6$	$(u\theta)_y = 56$	$(u\theta)_z = 137$

Comme précédemment l'algorithme de planification de trajectoire est initialisé en utilisant une estimation de la géométrie épipolaire. Les trois ensembles de paramètres désignés par calibration correcte, mauvaise calibration et très mauvaise calibration ont également été utilisés. La profondeur d^f a été prise respectivement égale à 70cm, 50cm et 100cm dans les cas calibration correcte, mauvaise calibration et très mauvaise calibration. Dans cette partie, nous nous contentons de présenter les résultats car leur description est identique à celle des cas où la cible était plane et inconnue. Notons cependant que les trajectoires obtenues, à la fois de la caméra dans l'espace cartésien et de l'objet dans l'image, sont particulièrement satisfaisantes dans les trois cas.

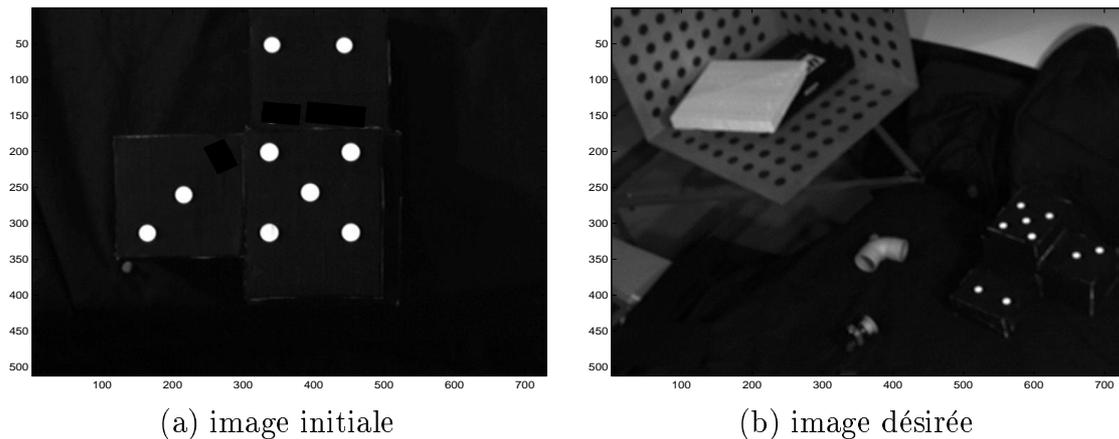
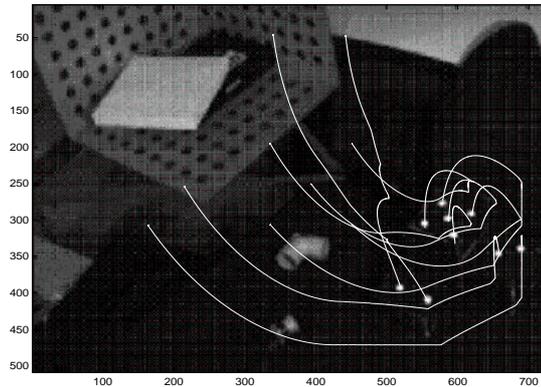
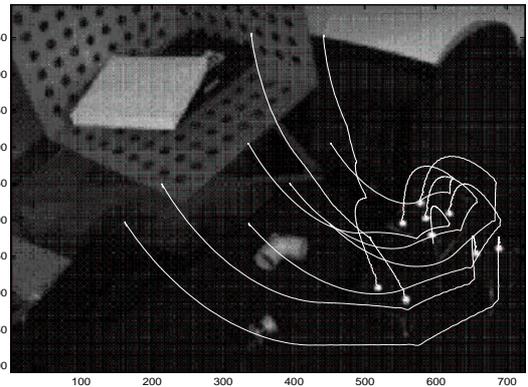


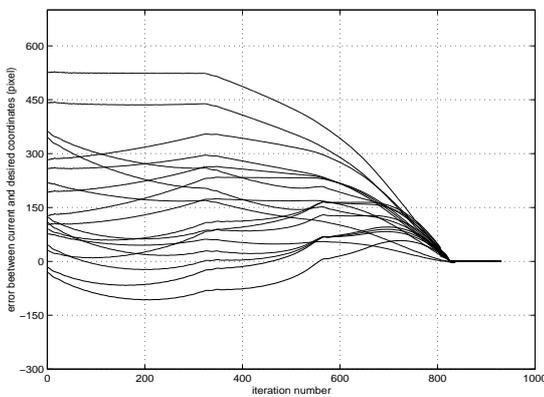
FIG. 6.14 – Images de la cible non-planaire



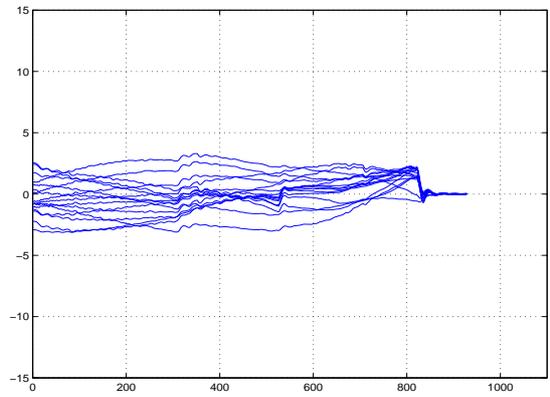
(a) trajectoires planifiées



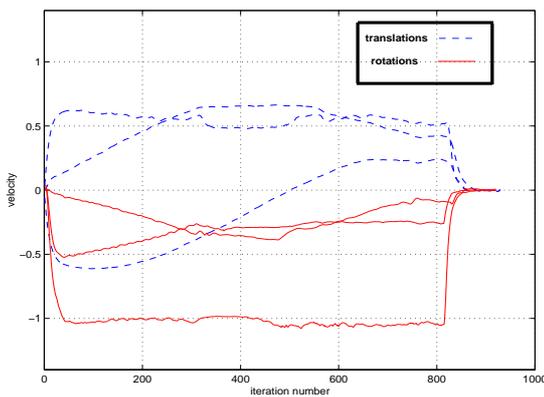
(b) trajectoires suivies



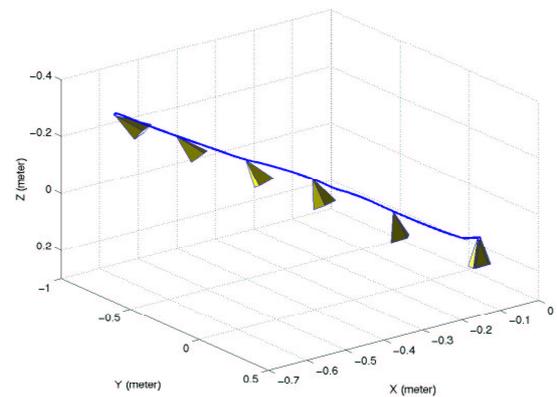
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel



(e) vitesse de la caméra (cm/s et deg/s)



(f) trajectoire de la caméra

FIG. 6.15 – Résultat dans le cas où le modèle de l'objet est inconnu et pour une calibration correcte

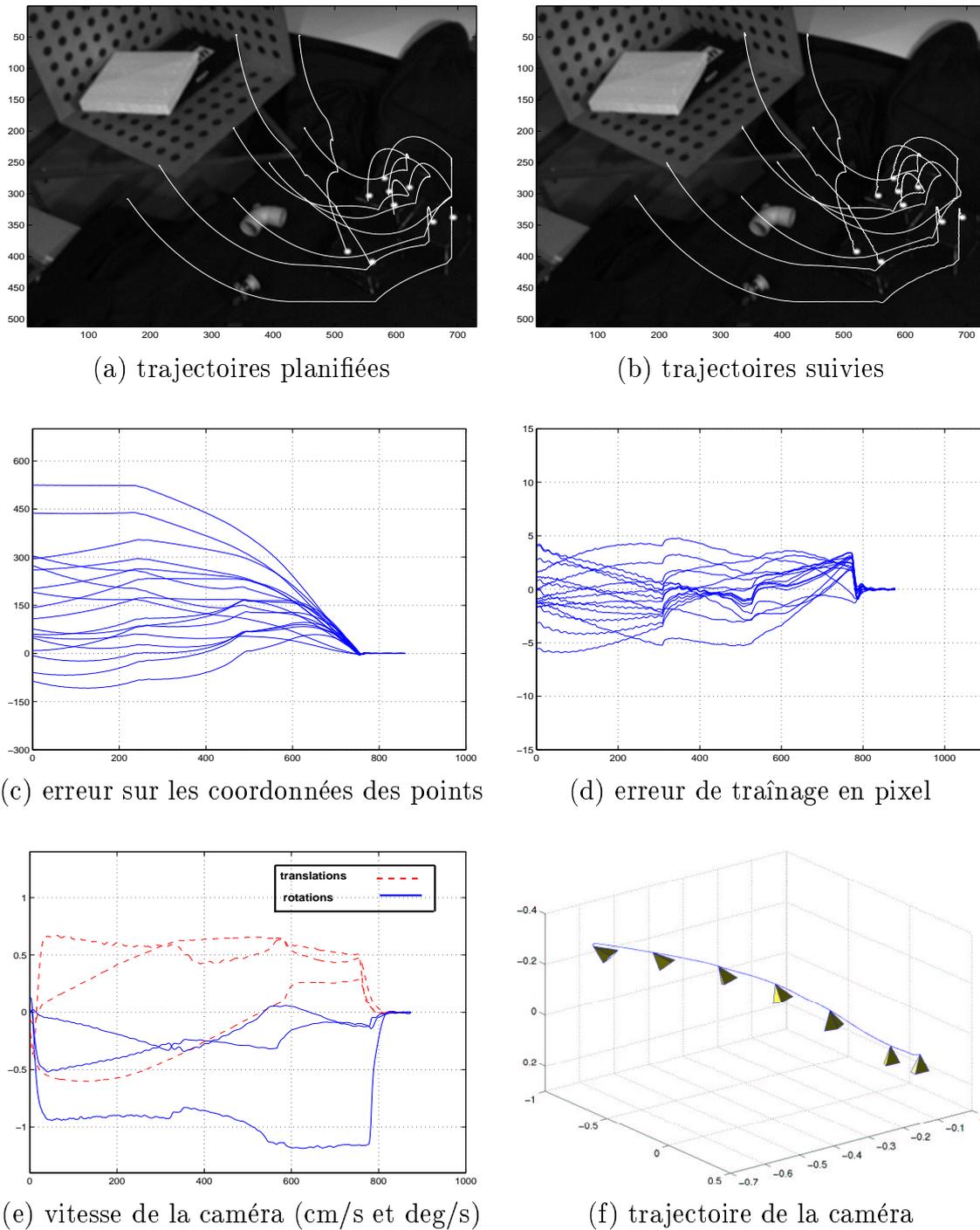
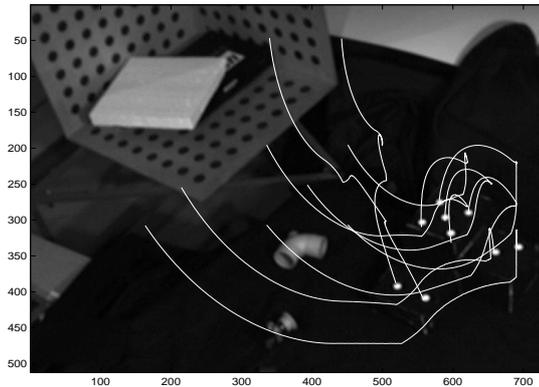
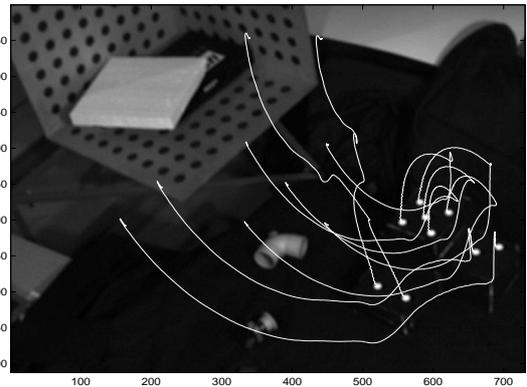


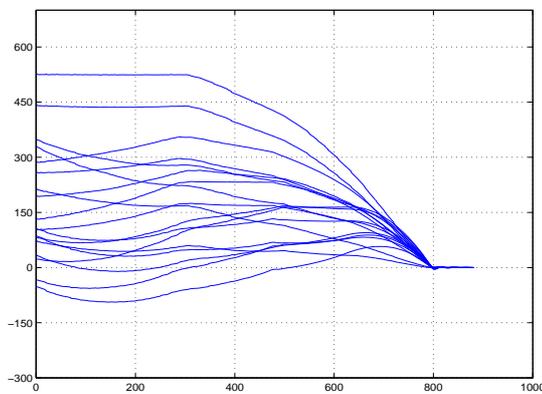
FIG. 6.16 – Résultat dans le cas où le modèle de l'objet est inconnu et pour une mauvaise calibration



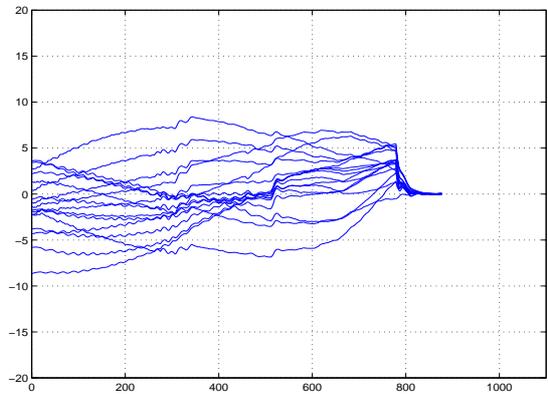
(a) trajectoires planifiées



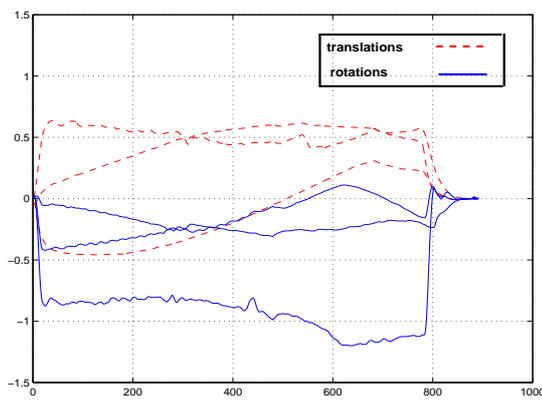
(b) trajectoires suivies



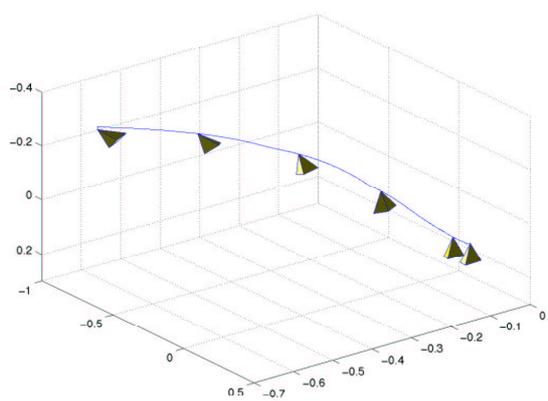
(c) erreur sur les coordonnées des points



(d) erreur de traînage en pixel



(e) vitesse de la caméra (cm/s et deg/s)



(f) trajectoire de la caméra

FIG. 6.17 – Résultat dans le cas où le modèle de l'objet est inconnu et pour une très mauvaise calibration

6.6.5 Evitement des butées articulaires

Dans cette partie, nous présentons deux expériences. Dans les deux cas, le modèle de l'objet est supposé inconnu. Dans la première expérience, la profondeur d^f et la calibration de la caméra sont correctes. Dans la seconde expérimentation, ces paramètres ne sont connus qu'approximativement. Les cibles utilisées sont composées de disques blancs sur fond noir. Les images initiale et désirée sont présentées sur les figures 6.18 et 6.21. Dans les figures relatives à cette section, les coordonnées articulaires sont normalisées entre $[-1;1]$, où -1 et 1 représentent les limites sur chaque axe. Afin de mettre en évidence l'importance des contraintes de visibilité et d'évitement des butées articulaires pour la réalisation de la tâche de positionnement, nous avons, dans un premier temps, utilisé la méthode de planification sans activer les forces répulsives. Nous remarquons que la tâche ne peut pas être réalisée car les primitives visuelles sortent largement du champ de vision de la caméra (voir figures 6.19(a) et 6.22(a)) et une des articulations du robot dépasse sa butée mécanique (voir figures 6.19(b) et 6.22(b)). Nous réalisons ensuite la même expérience en activant les forces répulsives. Dans ce cas, l'objet d'intérêt reste dans le champ de vision de la caméra, et aucune des butées articulaires n'est atteinte. La tâche peut alors être réalisée. La tâche est en effet correctement assurée, ce qui se vérifie en notant la convergence vers 0 de l'erreur des coordonnées dans l'image de tous les points considérés (voir figures 6.20(f) et 6.23(f)).

Dans la première expérience, les trajectoires réalisées sont très proches des trajectoires planifiées. Dans la seconde expérience, les trajectoires suivies dans l'espace articulaire ne correspondent pas parfaitement aux trajectoires planifiées. Ce comportement est essentiellement dû à l'utilisation de l'équation (6.20), qui est une approximation pour estimer la valeur courante des coordonnées articulaires. Ces erreurs d'approximation s'accumulent au cours du temps. L'estimation est donc d'autant plus mauvaise que le déplacement à effectuer est important. D'autre part, comme nous l'avons déjà précisé, la précision de la modélisation (caméra et profondeur) influence également la qualité de l'estimation de la trajectoire dans l'espace articulaire. Notons cependant, qu'en pratique, il est possible de prendre une marge de sécurité importante par rapport aux butées mécaniques en utilisant la distance d'influence des butées articulaires l^j (voir figure 6.2).

Pour pallier cette difficulté, il est possible d'introduire la contrainte liée aux butées articulaires dans un schéma de planification de trajectoires en ligne. Notons que la stratégie de planification adoptée peut s'adapter aisément à une planification de trajectoires en ligne. En procédant ainsi, l'estimation de la trajectoire articulaire n'est plus nécessaire car les coordonnées articulaires peuvent être mesurées à chaque itération de la boucle de commande. Cependant, l'échec de cette stratégie (par exemple en raison de la rencontre d'un minimum local) est plus problématique que dans un schéma de planification hors ligne. En effet, dans le premier cas, cela nécessite à la fois une nouvelle réalisation de la tâche et un changement de stratégie de planification alors que, dans le second cas, cela nécessite uniquement un change-

ment de stratégie de planification de trajectoires.

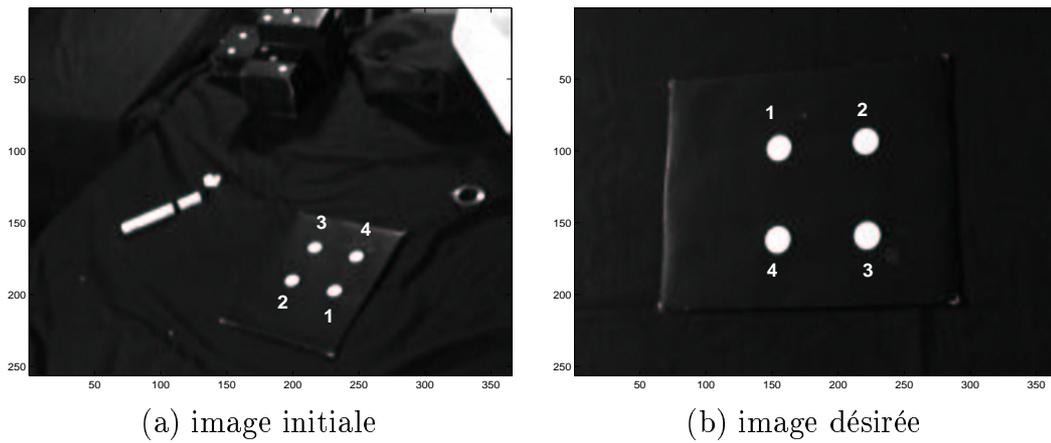


FIG. 6.18 – Image de la cible

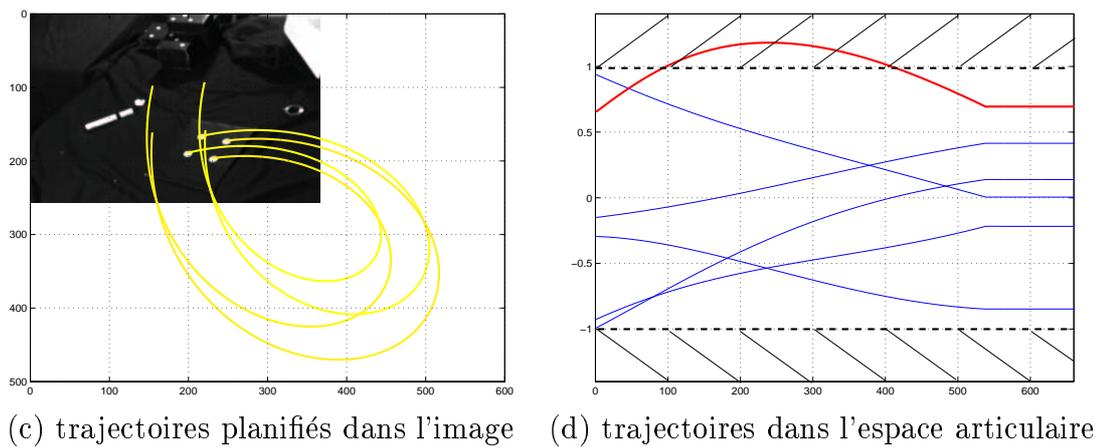
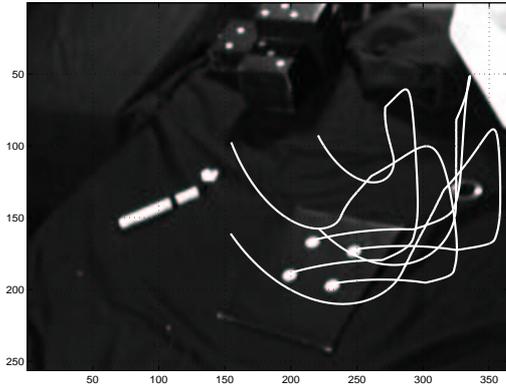
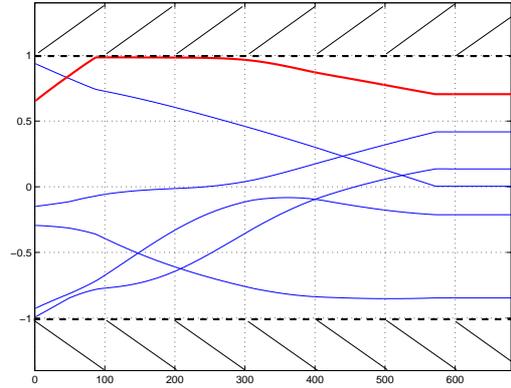


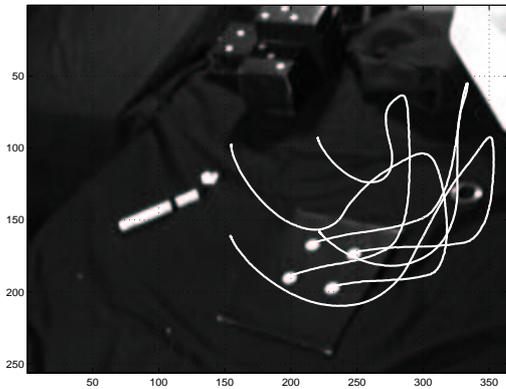
FIG. 6.19 – Trajectoire quand la force artificielle relative à l'évitement des butées articulaires n'est pas activée



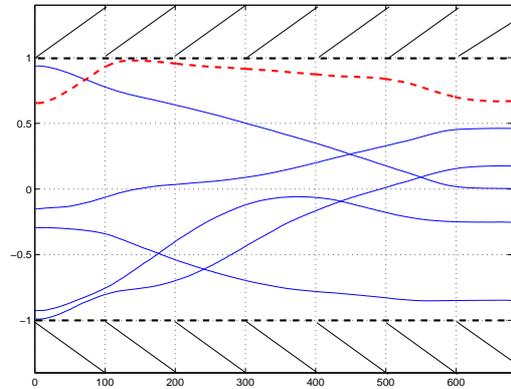
(a) trajectoires planifiées dans l'image



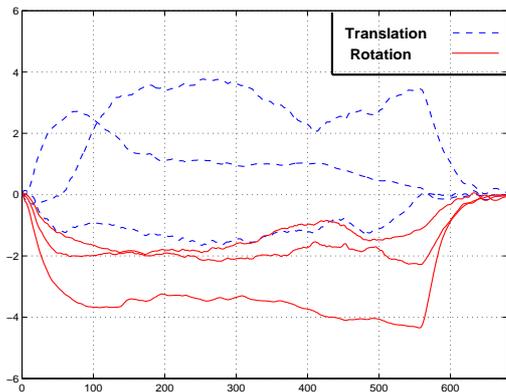
(b) trajectoires planifiées dans l'espace articulaire



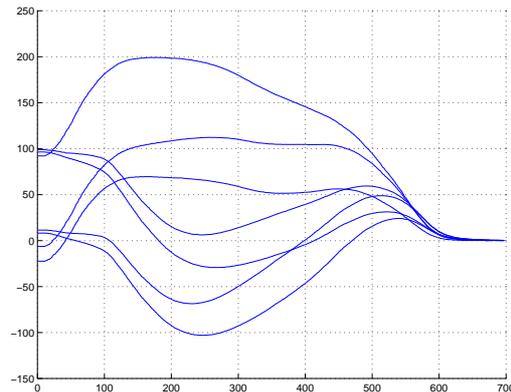
(c) trajectoires suivies dans l'image



(d) trajectoires suivies dans l'espace articulaire



(e) vitesse de la caméra (cm/s et deg/s)



(f) erreur sur les coordonnées des points en pixel

FIG. 6.20 – Les forces artificielles associées aux contraintes de visibilité et d'évitement des butées articulaires sont activées

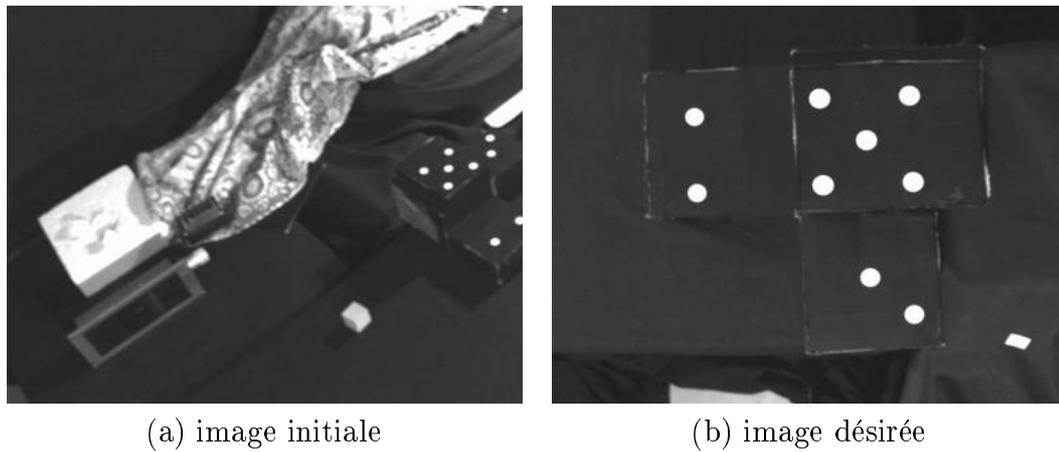


FIG. 6.21 – *Image de la cible*

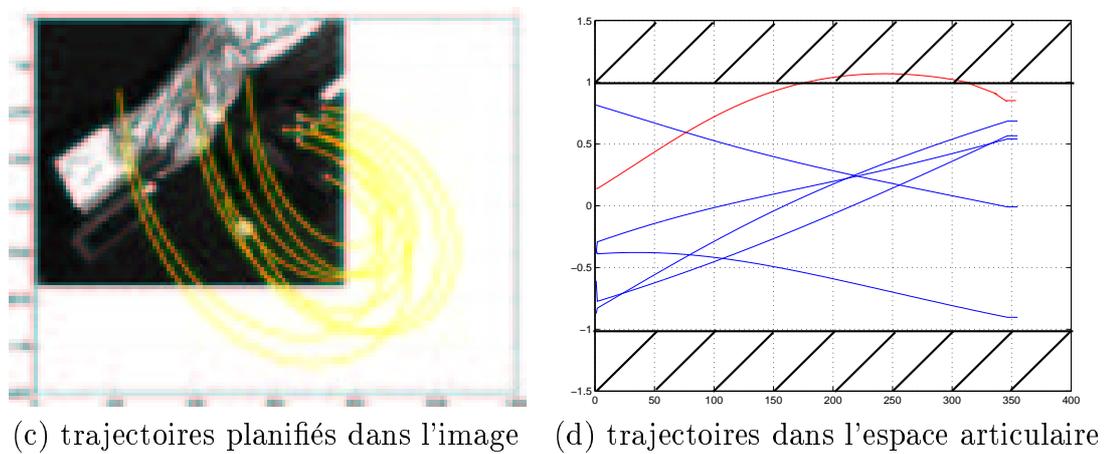
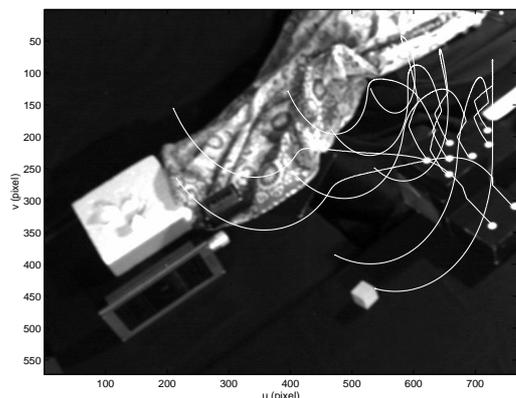
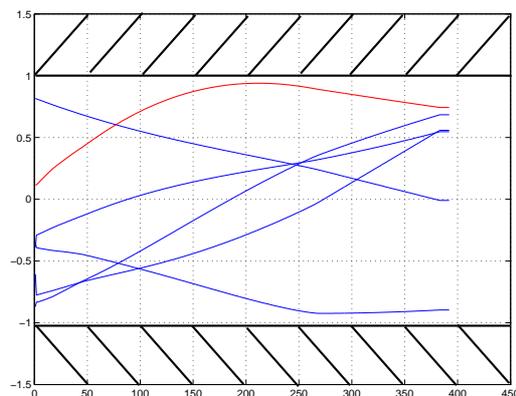


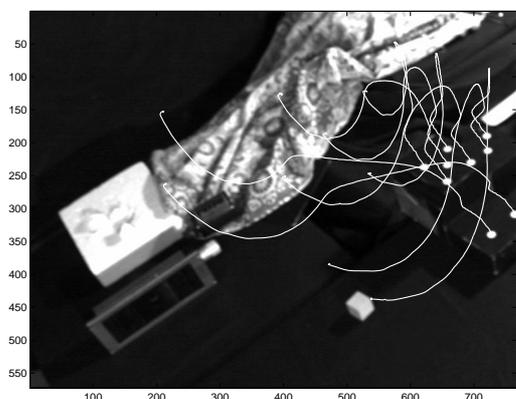
FIG. 6.22 – *Trajectoire quand la force artificielle relative à l'évitement des butées articulaires n'est pas activée*



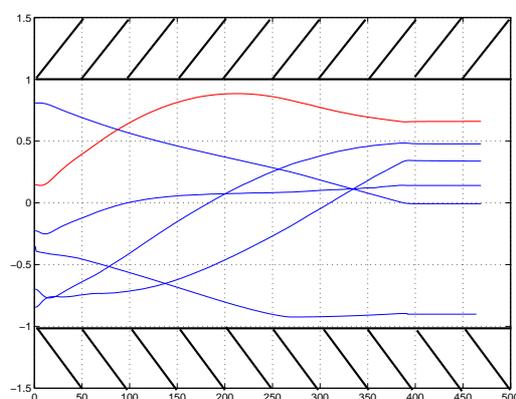
(a) trajectoires planifiées dans l'image



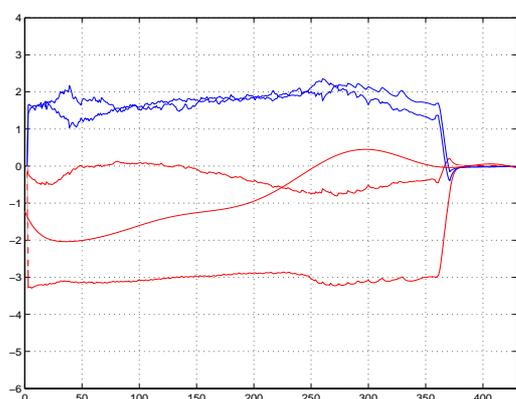
(b) trajectoires planifiées dans l'espace articulaire



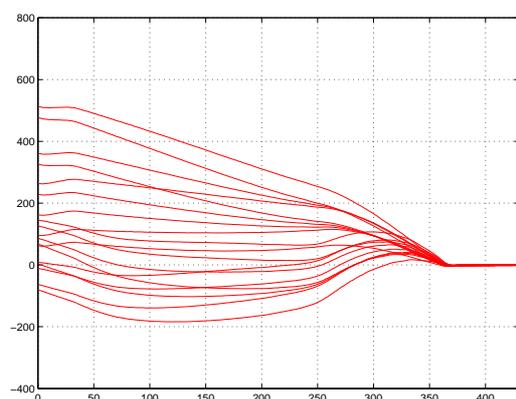
(c) trajectoires suivies dans l'image



(d) trajectoires suivies dans l'espace articulaire



(e) vitesse de la caméra (cm/s et deg/s)



(f) erreur sur les coordonnées des points en pixel

FIG. 6.23 – Les forces artificielles associées aux contraintes de visibilité et d'évitement des butées articulaires sont activées

6.7 Conclusion

Dans ce chapitre, nous avons proposé une méthode de planification de trajectoires dans l'espace de tâche, basée sur la méthode des potentiels.

Nous avons traité, dans un premier temps, le cas où le modèle de l'objet est disponible en utilisant le modèle sténopé pour la caméra. Nous avons ensuite vu que cette méthode pouvait être appliquée à une scène inconnue. Dans ce cas, la phase de planification nécessite uniquement la reconstruction projective puis euclidienne du déplacement partiel de la caméra entre les images initiale et désirée. Nous avons également étudié l'influence des erreurs de calibration et constaté leur faible influence sur les trajectoires obtenues si uniquement la contrainte de visibilité est prise en compte.

Par ailleurs, les trajectoires obtenues possèdent des propriétés très intéressantes. Le long de ces trajectoires, toute la cible reste dans le champ de vision de la caméra, les mouvements correspondant de la caméra sont physiquement réalisables et le centre optique passe de sa position initiale à sa position désirée le long d'une droite quand les forces de répulsions ne sont pas nécessaires. En annexe C, nous avons également traité le cas des auto-occultations.

En utilisant un asservissement visuel dans l'image afin de suivre les trajectoires planifiées dans l'espace de tâche, nous avons étendu le bon comportement (robustesse et stabilité) des asservissements $2D$ lorsque les positions initiale et désirée sont proches, aux cas où les positions initiale et désirée sont distantes.

Nous avons validé cette méthode sur des tâches de positionnement en utilisant un robot cartésien à six degrés de liberté. Les résultats obtenus sont très satisfaisants. En effet, nous avons pu réaliser des tâches de positionnement pour des déplacements et des erreurs de calibration très importants.

Conclusion

La tâche de base en robotique consiste à amener le robot d'une configuration initiale donnée à une configuration finale désirée. En général, le nombre de chemins possibles pour réaliser cette tâche est important. Cela reste vrai dans le contexte de l'asservissement visuel. Quel que soit le schéma de commande, l'utilisation d'une consigne fixe ne permet pas de sélectionner un chemin qui permette une réalisation satisfaisante de la tâche. Ce point est primordial car il est à l'origine de la totalité des échecs lorsque la tâche est réalisable. Par exemple, en utilisant une technique 3D, l'objet d'intérêt peut sortir de l'image, alors qu'un asservissement 2D aurait permis la réalisation de la tâche. A contrario, il peut s'avérer qu'une technique 2D conduise à une trajectoire irréalisable de l'effecteur du robot, alors qu'un asservissement 3D aurait permis d'obtenir une trajectoire très satisfaisante. Clairement, la sélection de la trajectoire à suivre parmi l'ensemble des trajectoires possibles est la solution à apporter. En outre, le schéma qui consiste à planifier une trajectoire puis à la suivre est très séduisant du point de vue de la commande, d'autant plus qu'on dispose de structures de commande connues pour être localement stables et robustes.

Le travail que nous avons présenté dans ce mémoire de thèse traite du choix d'un chemin satisfaisant pour la réalisation d'une tâche robotique lorsque les mesures sont fournies par un capteur de vision. Ce travail apporte en particulier une contribution à l'élargissement du champ d'application des techniques d'asservissement visuel 2D aux grands déplacements, cadre dans lequel la planification de trajectoires est nécessaire. Nous nous sommes efforcés de repousser les limites des techniques 2D dues à leur aspect local. Nos principales contributions au domaine de l'asservissement visuel sont détaillées ci-dessous.

Contributions

Nous avons recherché des méthodes permettant la réalisation de tâches robotiques en utilisant un capteur de vision, lorsque les déplacements à effectuer sont très importants. Il est apparu qu'une solution possible était le couplage d'une phase de planification de trajectoires dans l'espace du capteur et d'une phase de régulation basée sur les techniques 2D. Nous avons, dans ce cadre, proposé deux schémas de

planification :

- la première méthode repose sur le formalisme de la commande optimale,
- le principe du second schéma de planification de trajectoires dans l'image est basé sur la méthode des fonctions de potentiel.

Nous avons ensuite adapté les techniques classiques d'asservissement 2D à notre problématique, c'est-à-dire au suivi de trajectoires.

Plus précisément, en ce qui concerne les schémas de planification :

- pour la première méthode, nous avons obtenu les formes analytiques des trajectoires des primitives visuelles dans l'image. Les trajectoires dans l'image correspondent à un déplacement optimal de la caméra dans l'espace de travail. Nous avons considéré les problèmes à minimum d'énergie et minimum d'accélération. Nous avons également étudié le cas de l'interpolation d'un nombre quelconque d'images afin de pallier le problème de mise en correspondance des primitives visuelles lorsque les prises de vue initiale et finale sont distantes. Nous avons ensuite utilisé une mise en forme variationnelle du problème de planification de trajectoires dans l'image afin d'obtenir une solution numérique quand la contrainte de visibilité est prise en compte.
- pour la seconde méthode, nous avons traité, dans un premier temps, le cas où le modèle de l'objet est disponible en utilisant le modèle sténopé pour la caméra. Nous avons ensuite vu que cette méthode pouvait être appliquée à une scène inconnue. Par ailleurs, les trajectoires obtenues possèdent des propriétés très intéressantes. Le long de ces trajectoires, toute la cible reste dans le champ de vision de la caméra, les mouvements correspondant de la caméra sont physiquement réalisables et le centre optique passe de sa position initiale à sa position désirée le long d'une droite quand les forces de répulsions ne sont pas nécessaires.

En outre, pour les deux techniques proposées :

- les trajectoires dans l'image sont peu influencées par la calibration de la caméra,
- les modèles des objets observés par la caméra ne sont pas nécessaires

Par ailleurs, en utilisant l'asservissement visuel dans l'image afin de suivre les trajectoires planifiées dans l'espace de tâche, nous avons étendu le bon comportement des asservissements 2D, lorsque les positions initiale et désirée sont proches (robustesse et stabilité), aux cas où les positions initiale et désirée sont distantes.

Nous avons appliqué toutes les méthodes proposées à des tâches de positionnement de l'effecteur d'un robot manipulateur à six degrés de liberté. Nous avons pu réaliser, en utilisant nos méthodes, des déplacements très importants du robot. Les résultats que nous avons obtenus confirment l'amélioration importante apportée par

la phase de planification de trajectoires dans l'image et la robustesse des techniques proposées.

Perspectives

En ce qui concerne les méthodes de planification, nous avons vu que l'intégration de contraintes dans la méthode continue (et globale) est assez lourde en raison de la taille des systèmes d'équations non linéaires à résoudre et de l'étape d'initialisation. La seconde méthode, basée sur la méthode des potentiels, quant à elle, réalise une optimisation locale, ce qui permet une prise en compte aisée des contraintes. En contrepartie, les trajectoires obtenues peuvent être sous-optimales. Il serait intéressant pour pallier les problèmes de complexité de la première méthode et de sous-optimalité de la seconde, de mettre en œuvre les techniques, bien connues, de la programmation dynamique afin de réaliser une optimisation globale en présence de contraintes. En effet, dans ce cadre, l'ajout de contraintes ne complexifie pas de manière rébarbative la résolution du problème de planification. En outre, la question de complétude pourrait éventuellement être traitée.

Une suite naturelle à notre travail serait de considérer des environnements encombrés et donc de traiter le problème d'évitement d'obstacles statiques ou mobiles. Une étape de localisation dans l'image des éventuels obstacles serait alors à coupler à une stratégie de planification.

La planification de trajectoires pour les robots non-holonomes est en général réalisée dans l'espace de travail ou dans l'espace de ses actionneurs. L'intégration des contraintes, spécifiques à ce type de robot, directement dans un processus de planification dans l'espace du capteur (i.e dans l'espace image), nous semble également une perspective de recherche très prometteuse.

Dans le cadre de la robotique mobile, les tâches à réaliser nécessitent de très grands déplacements. Le problème de la mise en correspondance de primitives dans les images initiale et désirée devient très complexe, voire impossible si aucune primitive commune aux deux images ne peut être extraite. La solution qui nous semble la plus adaptée est l'utilisation d'images intermédiaires et acquises lors d'une phase d'apprentissage. Une base d'image pourrait être consultée et comparée aux images initiale et désirée afin d'en extraire les images intermédiaires nécessaires à la réalisation de la tâche. Dans cette optique, les techniques d'indexation d'images devrait pouvoir apporter une solution aux problèmes posés. Dans ce contexte, une phase de planification de trajectoires directement dans l'espace du capteur permettrait de construire un schéma de commande global et d'éviter les décélérations et accélérations au passage d'une image intermédiaire. Ce travail a fait l'objet d'un sujet de stage de DEA et se poursuivra au sein de l'IRISA sous la forme d'un travail de thèse.

Enfin, les expressions analytiques des trajectoires des pixels d'une image obtenues dans le cadre non calibré et correspondant à des déplacements optimaux de la caméra peuvent permettre la construction d'un ensemble continu d'images entre deux images cibles obtenues par une caméra non calibrée et observant une scène inconnue. Cette application de notre travail pourrait apporter une contribution au domaine de la synthèse d'images.

Annexes

Annexe A

Description d'une méthode numérique de résolution d'un problème de commande optimale

A.1 Introduction

Le problème central du contrôle optimal est d'amener un système dynamique d'un état initial à un état final désiré le long d'une trajectoire minimisant une fonction de coût bien choisie. Ce problème a tout d'abord été étudié par Lagrange au dix-huitième siècle. Des contributions importantes dans ce domaine ont ensuite été apportées par Mayer et Bolza, au début du dix-neuvième siècle, et un peu plus tard par Bliss. Les deux travaux les plus importants formalisant cette approche et en établissant les bases, sont le principe d'optimalité de la programmation dynamique de Bellman [Bellman 57] et le principe du minimum de Pontryagin [Pontryagin 62]. En pratique, il est rapidement apparu qu'un grand nombre de problèmes réels ne pouvait pas être résolu analytiquement. Cependant, les avancées dans les domaines des technologies de l'informatique et du calcul numérique ont permis d'obtenir des solutions numériques pour de nombreux problèmes de commande optimale. Dans cette optique, la formulation de Bolza est très intéressante pour décrire un problème de commande optimale sous une forme variationnelle. Cette forme variationnelle peut ensuite être exploitée pour obtenir une solution numérique au problème original.

A.2 Approche variationnelle

Dans la suite de cette annexe, nous allons décrire le problème classique de la commande optimale. Nous utiliserons le principe du minimum de Pontryagin pour obtenir une trajectoire de primitives géométriques dans l'image, correspondant à un déplacement optimal et continu de la caméra dans l'espace de travail. Nous verrons également que le problème classique de commande optimale peut être mis sous une

forme variationnelle équivalente, connue sous le nom de problème de Bolza.

A.2.1 La commande optimale

Considérons le système dynamique de dimension n décrit par l'équation différentielle du premier ordre suivante :

$$\dot{\mathbf{x}} = f(t, \mathbf{x}(t), \mathbf{u}(t)) \quad (\text{A.1})$$

où $\mathbf{x}(t) = [x_1(t) \ x_2(t) \ \cdots \ x_n(t)]^T$ et $\mathbf{u}(t) = [u_1(t) \ u_2(t) \ \cdots \ u_m(t)]^T$ représentent le vecteur d'état de dimension n et le vecteur de commande de dimension m . La fonction f est supposée suffisamment continue et dérivable, et le vecteur d'état est supposé appartenir à un sous-ensemble \mathcal{U} de \mathbb{R}^n . Si le système est non contraint, l'ensemble des commandes admissibles \mathcal{U}_a est l'ensemble des fonctions bornées, continues par morceaux. Considérons des fonctions suffisamment dérivables à valeur dans \mathbb{R} :

$$L(t, \mathbf{x}(t)) \quad \text{et} \quad \phi(\mathbf{x}) \quad (\text{A.2})$$

Nous définissons alors une fonction de coût, J , par la relation suivante :

$$J(\mathbf{x}, \mathbf{u}) = \phi(\mathbf{x}(t_0)) + \int_{t_0}^{t_f} L(t, \mathbf{x}(t), \mathbf{u}(t)) dt \quad (\text{A.3})$$

où t_0 et t_f représentent les instants initial et final.

Dans le cas où le système dynamique est contraint, l'ensemble des commandes admissibles \mathcal{U}_a sera considéré comme l'ensemble des fonctions bornées, continues par morceaux, satisfaisant l'ensemble des contraintes égalités et inégalités suivantes :

$$eq_i(t, \mathbf{x}, \mathbf{u}) = 0 \quad i = 1 \cdots n_{eq} \quad (\text{A.4})$$

et

$$in_i(t, \mathbf{x}, \mathbf{u}) \leq 0 \quad i = 1 \cdots n_{in} \quad (\text{A.5})$$

Le problème de commande optimale peut alors être formulé de la façon suivante :

Le problème de commande optimale *Etant donné le système dynamique décrit par l'équation (A.1), un état initial $\mathbf{x}_0 = \mathbf{x}(t_0)$ à un instant initial t_0 et un état final \mathbf{x}_f , alors trouver parmi tous les vecteurs de commande appartenant à \mathcal{U}_a la commande $\mathbf{u}^*(t)$ et le vecteur d'état correspondant $\mathbf{x}^*(t)$ qui évolue de l'état initial \mathbf{x}_0 à l'état final désiré \mathbf{x}_f et qui correspond au minimum de fonction de coût (ou de performance) J .*

Dans la suite, nous considérerons qu'il existe un ensemble ouvert de fonctions $\mathbf{u}(t)$ pouvant amener le système (A.1) de l'état initial à l'état désiré. Si ce n'est pas

le cas, alors cela n'a pas de sens de parler de la minimisation de la fonction de coût J . Quand cette hypothèse est satisfaite alors le problème est dit normal. La solution au problème fondamental de la commande optimale a été donnée par Pontryagin, et est connue sous le nom du principe du minimum de Pontryagin.

Principe du minimum de Pontryagin (problème non contraint) *Définissons l'hamiltonien :*

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{u}, t) \quad (\text{A.6})$$

Si la commande $\mathbf{u}^*(t)$ est optimale et génère une trajectoire du vecteur d'état $\mathbf{x}^*(t)$, alors il existe une solution non nulle $\boldsymbol{\lambda}^*(t)$ au système d'équations suivant :

$$\dot{\boldsymbol{\lambda}} = H_x = - \left(\frac{\partial H}{\partial \mathbf{x}} \right)^T \quad (\text{A.7})$$

telle que pour tout $t \in [t_0, t_f]$ et pour tout $\mathbf{u} \in \mathcal{U}_a$:

$$H(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, t) \leq H(\mathbf{x}^*, \mathbf{u}, \boldsymbol{\lambda}^*, t) \quad (\text{A.8})$$

Le vecteur $\boldsymbol{\lambda}(t)$ est appelé vecteur adjoint du système dynamique. Le principe du minimum de Pontryagin est très général et ne nécessite aucune hypothèse sur l'ensemble \mathcal{U}_a . Une formulation plus pratique de ce principe peut être obtenue si l'ensemble \mathcal{U}_a est un ensemble ouvert. Dans ce cas, la relation (A.8) peut se mettre sous la forme de la condition du première ordre d'optimalité suivante :

$$H_u = \frac{\partial H}{\partial \mathbf{u}} = 0 \quad (\text{A.9})$$

Notons également que la relation (A.1) peut être mise sous la forme :

$$H_\lambda = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \dot{\mathbf{x}} \quad (\text{A.10})$$

Les minima de la fonction de coût appartiennent à l'ensemble plus large de courbes rendant cette fonction de coût stationnaire. De telles courbes sont dites *extremales* et jouent un rôle central dans la théorie de la commande optimale et du calcul variationnel. Quand l'ensemble \mathcal{U}_a est ouvert, les points stationnaires de la fonction de coût sont exactement les courbes qui satisfont la relation (A.9). Dans ce cas, les extrema sont donnés par l'ensemble suivant d'équations :

$$\begin{cases} \dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} \\ \dot{\boldsymbol{\lambda}} = - \left(\frac{\partial H}{\partial \mathbf{x}} \right)^T \end{cases} \quad (\text{A.11})$$

où \mathbf{u} est obtenue à partir de (A.10). Si l'état initial \mathbf{x}_0 et l'état final \mathbf{x}_f sont donnés, alors ils fournissent toutes les conditions aux bornes nécessaires pour le système (A.7). Si la composante k , notée $(\mathbf{x}_0)_k$, du vecteur d'état initial n'est pas connue, elle doit être remplacée par [Sagan 69] :

$$(\mathbf{x}_0)_k = 0 \quad (\text{A.12})$$

De la même façon, si la composante $(\mathbf{x}_f)_k$ du vecteur d'état final n'est pas connue, elle doit être remplacée par la condition aux limites :

$$(\mathbf{x}_0)_k = \frac{\partial \phi}{\partial (\mathbf{x})_k} \quad (\text{A.13})$$

Les relations (A.12) et (A.13) sont appelées *conditions aux limites naturelles* et sont des cas particuliers des *conditions de transversalité*. Supposons maintenant que l'ensemble des commandes admissibles soit l'ensemble des fonctions bornées continues par morceaux vérifiant l'ensemble des contraintes égalités et inégalités données par les équations (A.4) et (A.5), alors le principe du minimum peut être formulé en incorporant les contraintes directement dans l'Hamiltonien.

Principe du minimum de Pontryagin (problème contraint) *Définissons l'Hamiltonien par :*

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\mu}^T \mathbf{eq}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\nu}^T \mathbf{in}(\mathbf{x}, \mathbf{u}, t) \quad (\text{A.14})$$

où $\boldsymbol{\mu}$ et $\boldsymbol{\nu}$ sont respectivement des vecteurs de dimension n_{eq} et n_{in} et :

$$\mathbf{eq} = [eq_1, eq_2, \dots, eq_{n_{eq}}]^T \quad (\text{A.15})$$

et :

$$\mathbf{in} = [in_1, in_2, \dots, in_{n_{in}}]^T \quad (\text{A.16})$$

Si la commande $\mathbf{u}^*(t)$ est optimale et génère une trajectoire $\mathbf{x}^*(t)$, alors il existe une solution non nulle $\boldsymbol{\lambda}^*(t)$ au système adjoint :

$$\dot{\boldsymbol{\lambda}} = H_x = - \left(\frac{\partial H}{\partial \mathbf{x}} \right)^T \quad (\text{A.17})$$

et la commande optimale \mathbf{u}^* vérifie la relation :

$$H_u = \frac{\partial H(\mathbf{x}, \mathbf{u}, \boldsymbol{\mu}, \boldsymbol{\nu}, t)}{\partial \mathbf{u}} = 0 \quad (\text{A.18})$$

et les conditions supplémentaires suivantes doivent être satisfaites :

$$\begin{cases} \nu_i = 0 & \text{if } in_i < 0 \\ \nu_i \geq 0 & \text{if } in_i = 0 \end{cases} \quad (\text{A.19})$$

A.2.2 Le problème de Bolza

Une approche très intéressante pour résoudre le problème de la commande optimale est de le convertir en un problème de calcul variationnel connu sous le nom de problème de Bolza [Berkovitz 61]. Formellement, le problème de Bolza est défini de la manière suivante :

Problème de Bolza. *Considérons des fonctions suffisamment dérivables $\bar{\Psi}$ et \bar{L} à valeur dans \mathfrak{R} , alors trouver une fonction continue par morceaux $\mathbf{X}(t)$ de dimension r qui minimise la fonctionnelle :*

$$J = \bar{\Psi}(\mathbf{X}(t_0), t_0) + \int_{t_0}^{t_f} \bar{L}(\mathbf{X}, \dot{\mathbf{X}}, t) dt \quad (\text{A.20})$$

sous les contraintes :

$$\phi_i(\mathbf{X}, \dot{\mathbf{X}}, t) = 0 \quad i = 1 \cdots p \quad (p \leq r), \quad (\text{A.21})$$

vérifiant la condition initiale :

$$\mathbf{X}(t_0) = \mathbf{X}_0 \quad (\text{A.22})$$

et la condition finale :

$$\mathbf{X}(t_f) = \mathbf{X}_f \quad (\text{A.23})$$

Une nouvelle fois nous supposerons que le problème est normal, c'est-à-dire qu'il existe un ensemble ouvert de fonctions $\mathbf{X}(t)$ solutions du système d'équations (A.21)-(A.22)-(A.23). Le problème de la commande optimale peut être reformulé sous la forme d'un problème de Bolza en utilisant la méthode proposée par Valentine [Valentine 37] pour convertir des contraintes inégalités en contraintes égalités, et en introduisant des variables supplémentaires dans le vecteur de commande du système initial. Les conditions nécessaires sur les solutions du problème de Bolza peuvent être obtenues en utilisant les résultats classiques du calcul variationnel [Bliss 61] :

Proposition. Une fonction $z(t)$ est dite admissible si elle est continue par morceaux et telle que $z(t_0) = z(t_f) = 0$. Si une fonction continue par morceaux $\mathbf{X}(t)$ est solution du problème de Bolza, alors il existe un vecteur de dimension p non nul de multiplicateur de Lagrange $\mathbf{\Gamma}$ et une fonction H définie par :

$$H = \bar{L} + \mathbf{\Gamma}^T \boldsymbol{\phi} \quad (\text{A.24})$$

tels que :

$$\int_{t_0}^{t_f} \left(\frac{\partial H}{\partial \mathbf{X}} z + \frac{\partial H}{\partial \dot{\mathbf{X}}} \dot{z} \right) dt = 0 \quad (\text{A.25})$$

pour toute fonction admissible z . Les contraintes suivantes doivent également être satisfaites :

$$\phi = 0 \quad (\text{A.26})$$

Après quelques manipulations supplémentaires, nous pouvons obtenir les conditions du premier ordre pour la résolution du problème de Bolza connues sous les noms d'équations d'Euler-Lagrange et de conditions de Weierstrass-Erdmann.

Equations d'Euler-Lagrange. Si une fonction continue par morceaux $\mathbf{X}(t)$ est solution du problème de Bolza, alors il existe un vecteur de multiplicateurs de Lagrange Γ de dimension p et une fonction H définie par :

$$H = \bar{L} + \Gamma^T \phi \quad (\text{A.27})$$

tels qu' en tout point où le vecteur $\mathbf{X}(t)$ est continu, la relation d'Euler-Lagrange soit satisfaite :

$$\frac{d}{dt} \frac{\partial H}{\partial \dot{\mathbf{X}}} - \frac{\partial H}{\partial \mathbf{X}} = 0 \quad (\text{A.28})$$

et tels qu'en chaque point t de discontinuité, les conditions de Weierstrass-Erdmann soient vérifiées :

$$\begin{aligned} \left(\frac{\partial H}{\partial \dot{\mathbf{X}}} \right)_{t^-} &= \left(\frac{\partial H}{\partial \dot{\mathbf{X}}} \right)_{t^+} \\ \left(\dot{\mathbf{X}} \frac{\partial H}{\partial \dot{\mathbf{X}}} - H \right)_{t^-} &= \left(\dot{\mathbf{X}} \frac{\partial H}{\partial \dot{\mathbf{X}}} - H \right)_{t^+} \end{aligned} \quad (\text{A.29})$$

En outre, les contraintes suivantes doivent être satisfaites :

$$\phi = 0 \quad (\text{A.30})$$

A.2.3 Reformulation du problème de la commande optimale sous la forme de Bolza

La reformulation du problème de la commande optimale sous la forme d'un problème de Bolza est très intéressante car l'obtention d'une solution numérique pour le problème de la commande optimale est en général une tâche difficile. Nous formulerons le problème de la commande optimale sous contraintes, en un problème de calcul de variations non contraint pour lequel l'objectif est la minimisation de l'intégrale d'un Hamiltonien H :

$$J = \int_{t_0}^{t_f} H(\mathbf{X}(t), \dot{\mathbf{X}}(t), t) dt \quad (\text{A.31})$$

Dans nos applications, la variable temporelle n'intervenant pas de manière explicite nous ne la ferons plus apparaître pour simplifier les notations. Nous définissons l'Hamiltonien en y intégrant la fonction de coût et l'ensemble des contraintes. Les contraintes inégalités sont ajoutées à l'Hamiltonien après avoir été converties en contraintes égalités. En effet, chaque contrainte inégalité peut être convertie en une contrainte égalité équivalente en introduisant un vecteur de variable $\zeta = [\zeta_1 \cdots \zeta_{n_{in}}]^T$ de dimension n_{in} [Valentine 37] :

$$in_i(\mathbf{x}, \mathbf{u}) \leq 0 \Leftrightarrow \bar{i}n_i = in_i(\mathbf{x}, \mathbf{u}) + \zeta_i^2 = 0 \quad i = 1 \cdots n_{in} \quad (\text{A.32})$$

L'Hamiltonien est alors défini de la manière suivante :

$$H(\mathbf{X}, \dot{\mathbf{X}}) = L(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^T (\dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}, \mathbf{u})) + \boldsymbol{\mu}^T \mathbf{eq}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\nu}^T \bar{i}\mathbf{n}(\mathbf{x}, \mathbf{u}) \quad (\text{A.33})$$

Dans cette dernière relation, \mathbf{X} est le vecteur d'état étendu :

$$\mathbf{X} = [\mathbf{X}_x^T, \mathbf{X}_u^T, \mathbf{X}_\zeta^T, \mathbf{X}_\lambda^T, \mathbf{X}_\mu^T, \mathbf{X}_\nu^T]^T \quad (\text{A.34})$$

où $\mathbf{X}_x = \mathbf{x}$ représente le vecteur d'état du problème de commande optimale original, et :

$$\begin{aligned} \dot{\mathbf{X}}_u &= \mathbf{u}; & \dot{\mathbf{X}}_\zeta &= \boldsymbol{\zeta}; & \dot{\mathbf{X}}_\lambda &= \boldsymbol{\lambda}; \\ \dot{\mathbf{X}}_\mu &= \boldsymbol{\mu}; & \dot{\mathbf{X}}_\nu &= \boldsymbol{\nu}. \end{aligned} \quad (\text{A.35})$$

$\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ et $\boldsymbol{\nu}$ sont respectivement des multiplicateurs de Lagrange de dimensions n , n_{eq} et n_{in} . Le problème variationnel non contraint consiste alors à trouver $\mathbf{X}(t)$ minimisant la fonctionnelle (A.31) et vérifiant les conditions aux limites. Nous supposons que les vecteurs d'état initial $\mathbf{X}_x(t_0) = \mathbf{x}_0$ et final $\mathbf{X}_x(t_f) = \mathbf{x}_f$ correspondant au problème de commande optimale original sont connus. Les vecteurs $\dot{\mathbf{X}}_u$, $\dot{\mathbf{X}}_\zeta$, $\dot{\mathbf{X}}_\lambda$, $\dot{\mathbf{X}}_\mu$ et $\dot{\mathbf{X}}_\nu$ peuvent être choisis nuls à l'instant initial et doivent satisfaire les conditions de transversalité à l'instant final. Les conditions aux limites sont donc données par :

$$\begin{aligned} \mathbf{X}_x(t_0) &= \mathbf{x}_0; & \mathbf{X}_x(t_f) &= \mathbf{x}_f; & \dot{\mathbf{X}}_u(t_0) &= 0; & \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_u} \right) (t_f) &= 0; \\ \dot{\mathbf{X}}_\zeta(t_0) &= 0; & \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\zeta} \right) (t_f) &= 0; & \dot{\mathbf{X}}_\lambda(t_0) &= 0; & \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\lambda} \right) (t_f) &= 0; \\ \dot{\mathbf{X}}_\mu(t_0) &= 0; & \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\mu} \right) (t_f) &= 0; & \dot{\mathbf{X}}_\nu(t_0) &= 0; & \left(\frac{\partial H}{\partial \dot{\mathbf{X}}_\nu} \right) (t_f) &= 0 \end{aligned} \quad (\text{A.36})$$

Pour que $\mathbf{x}(t)$ et $\mathbf{u}(t)$ soient solutions du système d'équations (A.1)-(A.3)-(A.4)-(A.5), il faut qu'une fonction continue par morceaux $\mathbf{X}(t)$ soit solution de (A.28). La méthode dont nous avons décrit les grandes lignes dans cette partie, peut directement être utilisée telle quelle quand le problème posé ne fait intervenir que des considérations cinématiques. Quand la dynamique du système est importante,

le problème devient plus compliqué. Dans ce cas, l'intégration des contraintes exige une attention particulière, le lecteur intéressé par ce point pourra par exemple se reporter à [Jacobson 69]. Nous utilisons une formulation de Bolza pour des problèmes où les considérations sont purement cinématiques (trajectoires de plus courte distance). Nous pouvons donc directement utiliser la procédure décrite précédemment pour convertir un problème de commande optimale sous contraintes en un problème de calcul variationnel non contraint. Nous allons voir dans la section suivante comment obtenir une solution numérique au problème de Bolza.

A.2.4 Résolution numérique du problème de Bolza

Rappelons que si \mathbf{X} est une solution du problème de Bolza, alors pour toute fonction admissible $z(t)$ la relation suivante doit être vérifiée :

$$\int_{t_0}^{t_f} (\mathbf{H}_X z + \mathbf{H}_{\dot{X}} \dot{z}) dt = 0 \quad (\text{A.37})$$

où \mathbf{H}_X et $\mathbf{H}_{\dot{X}}$ représentent les dérivées partielles de H par rapport à \mathbf{X} et $\dot{\mathbf{X}}$. Classiquement, les extrema sont obtenus en résolvant les équations de Euler-Lagrange (A.28). Le principal inconvénient est que ces équations sont valables uniquement aux points où l'extremum est continu. Aux points de discontinuité, ce sont les conditions de Weirstrass-Erdmann (A.29) qui doivent être utilisées. Donc, la position, le nombre et l'amplitude des points de discontinuités doivent être connus a priori. Cela rend très difficile la construction d'une méthode numérique générale pour la résolution du problème de Bolza en utilisant les équations d'Euler-Lagrange. Une alternative a été suggérée par Gregory et Lin [Gregory 92]. Ils basent leur méthode de résolution numérique du problème de Bolza sur la forme intégrale de la condition nécessaire d'optimalité (A.37). En effet, la forme intégrale est valable en tout point. Pour obtenir une solution numérique, ils choisissent une fonction admissible $z(t)$ particulière. Dans un premier temps, l'horizon temporel $[t_0, t_f]$ est discrétisé de telle manière que $t_0 < t_1 < \dots < t_N = t_f$ et $t_k - t_{k-1} = h$ pour $k = 1 \dots N$. Ensuite, la fonction linéaire par morceaux suivante est définie (voir figure A.1) :

$$\gamma_k(t) = \begin{cases} \frac{t-t_{k-1}}{h} & \text{si } t_{k-1} < t \leq t_k, \\ \frac{t_{k+1}-t}{h} & \text{si } t_k < t \leq t_{k+1}, \\ 0 & \text{sinon} \end{cases} \quad k = 0 \dots N-1 \quad (\text{A.38})$$

Comme l'équation (A.37) est valable pour toute fonction $z(t)$ admissible, elle doit être vérifiée pour la fonction $z(t)$ choisie telle que sa composante k soit égale à γ_k , et telle que toutes les autres composantes soient nulles. Ensuite, en utilisant les différences finies comme estimation de la dérivée et le théorème de la moyenne

pour estimer l'intégrale (A.37) sur chaque intervalle $[t_{k-1}, t_k]$, on aboutit au système d'équations vectorielles pour les inconnues \mathbf{X}_k aux points $k = 1, \dots, N - 1$:

$$\frac{h}{2} \mathbf{H}_X(\mathbf{X}_{k-1}, \mathbf{X}_k) + \mathbf{H}_{\dot{X}}(\mathbf{X}_{k-1}, \mathbf{X}_k) + \frac{h}{2} \mathbf{H}_X(\mathbf{X}_k, \mathbf{X}_{k+1}) - \mathbf{H}_{\dot{X}}(\mathbf{X}_k, \mathbf{X}_{k+1}) = 0 \quad (\text{A.39})$$

Le vecteur \mathbf{X}_k est de dimension p , le système d'équations (A.39) comporte donc $p(N+1)$ inconnues. Ce système fournit également $p(N-1)$ équations. Les conditions aux bornes fournissent les $2p$ équations supplémentaires nécessaires au calcul des $p(N+1)$ inconnues. Le système complet de $p(N+1)$ équations non-linéaires peut être résolu en utilisant une méthode adaptée comme par exemple la méthode de Newton-Raphson. Chacune des équations du système (A.39) dépend uniquement de la valeur des variables inconnues en trois points successifs. Cette propriété est très intéressante car le système linéarisé qui est résolu dans la méthode de Newton-Raphson est bloc-tridiagonal et le coût en temps de calcul peut alors être réduit de manière significative.

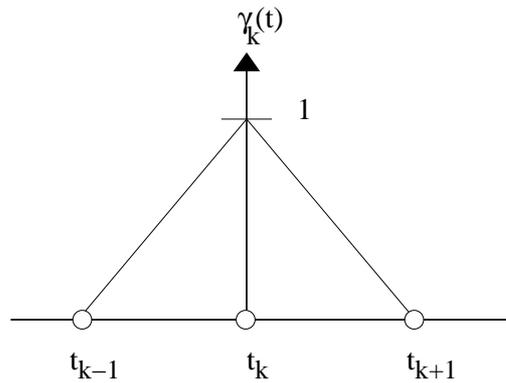


FIG. A.1 – Fonction utilisée dans la résolution numérique du problème de Bolza

Annexe B

Trajectoires optimales de la caméra

Dans cette annexe, nous cherchons les trajectoires optimales de la caméra correspondant aux problèmes *PCN1* et *PCN2*. Nous rappelons que ces problèmes se mettent sous les formes suivantes :

– problème correspondant à *PCN1*:

Trouver $\mathbf{R}(t)$ et $\mathbf{t}(t)$ minimisant $J_5 = \int_0^1 \mathbf{U}^T \mathbf{U} dt$
avec :

$$\mathbf{R}^T \dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times},$$

$$\mathbf{v} = \dot{\mathbf{t}}$$

et les conditions aux limites suivantes :

$$\mathbf{R}(0) = \mathbf{R}_0$$

$$\mathbf{R}(1) = \mathbf{I}$$

$$\mathbf{t}(0) = \mathbf{t}_0$$

$$\mathbf{t}(1) = \mathbf{0}$$

– problème correspondant à *PCN2*:

Trouver $\mathbf{R}(t)$ et $\mathbf{t}(t)$ minimisant $J_6 = \int_0^1 \dot{\mathbf{U}}^T \dot{\mathbf{U}} dt$
avec :

$$\begin{aligned}\mathbf{R}^T \dot{\mathbf{R}} &= [\boldsymbol{\omega}]_{\times}, \\ \mathbf{v} &= \dot{\mathbf{t}}\end{aligned}$$

et les conditions aux limites suivantes :

$$\begin{aligned}\mathbf{R}(0) &= \mathbf{R}_0 \\ \mathbf{R}(1) &= \mathbf{I} \\ \mathbf{t}(0) &= \mathbf{t}_0 \\ \mathbf{t}(1) &= \mathbf{0} \\ \mathbf{U}(0) &= \mathbf{U}(1) = \mathbf{0}\end{aligned}$$

B.1 Problème PCN1

Dans la suite, nous notons $\boldsymbol{\theta}$ le vecteur de dimension 3 tel que $\dot{\boldsymbol{\theta}} = \boldsymbol{\omega}$. Considérons l'état du système défini de la manière suivante :

$$\mathbf{X}(t) = \begin{bmatrix} \mathbf{t} \\ \boldsymbol{\theta} \end{bmatrix}$$

avec $\mathbf{X}(t_0) = \mathbf{X}_0$, $\mathbf{X}(t_1) = \mathbf{X}_1$. L'équation d'état est donnée par :

$$\dot{\mathbf{X}} = \mathbf{U} \tag{B.1}$$

La fonction de coût est :

$$J = \int_0^1 \mathbf{U}^T \mathbf{U} dt \tag{B.2}$$

Le problème de commande optimale est alors de trouver, parmi l'ensemble des commandes admissibles, la commande qui amène le système (B.1) de l'état initial \mathbf{X}_0 à l'état désiré \mathbf{X}_1 et minimise la fonction de coût (B.2). L'Hamiltonien est donné par :

$$H = \mathbf{U}^T \mathbf{U} + \boldsymbol{\lambda}^T \mathbf{U} \tag{B.3}$$

où $\boldsymbol{\lambda}$ représente le vecteur des co-états de dimension 6. L'équation des co-états est donnée par (voir (A.11)) :

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{X}} = \mathbf{0} \tag{B.4}$$

et la condition de stationnarité s'écrit (se référer à (A.9)) :

$$\frac{\partial H}{\partial \mathbf{U}} = \boldsymbol{\lambda} + 2\mathbf{U} = \mathbf{0}_{6 \times 1} \quad (\text{B.5})$$

D'après les relations (B.4) et (B.5) :

$$\mathbf{U} = -\frac{\boldsymbol{\lambda}}{2} = \mathbf{c} \quad (\text{B.6})$$

où \mathbf{c} est un vecteur de dimension 6 constant. En utilisant ensuite l'équation d'état, la relation (5.5) et les conditions aux limites, on aboutit aisément à :

$$\begin{cases} \mathbf{t}(t) = (1-t)\mathbf{t}_0 \\ \mathbf{R}(t) = \mathbf{R}_0 \mathbf{e}^{[\mathbf{r}_0] \times t} \end{cases} \quad (\text{B.7})$$

où $[\mathbf{r}_0]_{\times} = \log(\mathbf{R}_0^T)$.

B.2 Problème PCN2

Nous choisissons maintenant le vecteur d'état comme le vecteur de dimension 12 suivant :

$$\mathbf{X}(t) = \begin{bmatrix} \mathbf{t} \\ \mathbf{v} \\ \boldsymbol{\theta} \\ \boldsymbol{\omega} \end{bmatrix}$$

avec les conditions aux limites suivantes $\mathbf{X}(t_0) = \mathbf{X}_0$, $\mathbf{X}(t_1) = \mathbf{X}_1$ et le vecteur de commande comme le vecteur de dimension 6 suivant : $\mathbf{U} = [\dot{\mathbf{v}}^T \dot{\boldsymbol{\omega}}^T]^T$. L'équation d'état est donc :

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U} \quad (\text{B.8})$$

où la matrice de dimension 12×12 , \mathbf{A} , et la matrice de dimension 12×6 , \mathbf{B} sont données par :

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (\text{B.9})$$

où $\mathbf{0}$ et \mathbf{I} représentent les matrices nulle et identité de dimension 3×3 . La fonction de coût à minimiser est :

$$J = \int_0^1 \mathbf{U}^T \mathbf{U} dt \quad (\text{B.10})$$

Le problème de commande optimale est alors de trouver, parmi l'ensemble des commandes admissibles, la commande qui amène le système (B.8) de l'état initial \mathbf{X}_0 à l'état désiré \mathbf{X}_1 et minimise la fonction de coût (B.11). L'Hamiltonien est donné par :

$$H = \mathbf{U}^T \mathbf{U} + \boldsymbol{\lambda}^T (\mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}) \quad (\text{B.11})$$

où $\boldsymbol{\lambda}$ est le vecteur des co-états de dimension 12. L'équation des co-états s'écrit :

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{X}} = -\mathbf{A}^T \boldsymbol{\lambda} \quad (\text{B.12})$$

et la condition de stationnarité est donnée par :

$$\frac{\partial H}{\partial \mathbf{U}} = \mathbf{B}^T \boldsymbol{\lambda} + 2\mathbf{U} = \mathbf{0}_{6 \times 1} \quad (\text{B.13})$$

A partir de (B.12), nous obtenons :

$$\begin{cases} [\lambda_1 & \lambda_2 & \lambda_3 & \lambda_7 & \lambda_8 & \lambda_9]^T = \mathbf{c}_0^0 \\ [\lambda_4 & \lambda_5 & \lambda_6 & \lambda_{10} & \lambda_{11} & \lambda_{12}]^T = \mathbf{c}_0^0 t + \mathbf{c}_1^1 \end{cases} \quad (\text{B.14})$$

où \mathbf{c}_0^0 et \mathbf{c}_1^1 sont deux vecteurs constants de dimension 6. En introduisant (B.14) dans (B.13), on aboutit à :

$$\mathbf{U} = \mathbf{c}_0 t + \mathbf{c}_1 \quad (\text{B.15})$$

avec $\mathbf{c}_0 = -\frac{1}{2}\mathbf{c}_0^0$ et $\mathbf{c}_1 = -\frac{1}{2}\mathbf{c}_1^1$. En intégrant la relation (B.15), on arrive à :

$$\begin{cases} \mathbf{v} = \frac{1}{2}\mathbf{c}_{0v}t^2 + \mathbf{c}_{1v}t + \mathbf{c}_{2v} \\ \boldsymbol{\omega} = \frac{1}{2}\mathbf{c}_{0\omega}t^2 + \mathbf{c}_{1\omega}t + \mathbf{c}_{2\omega} \end{cases} \quad (\text{B.16})$$

Puis en utilisant les conditions aux limites, $\mathbf{v}(0) = \mathbf{v}(1) = \boldsymbol{\omega}(0) = \boldsymbol{\omega}(1) = \mathbf{0}$, nous concluons que $\mathbf{c}_{2v} = \mathbf{c}_{2\omega} = \mathbf{0}$ et $\mathbf{c}_{1v} = -\frac{1}{2}\mathbf{c}_{0v}$, $\mathbf{c}_{1\omega} = -\frac{1}{2}\mathbf{c}_{0\omega}$. L'équation (B.16) peut donc être mise sous la forme suivante :

$$\begin{cases} \mathbf{v} = \frac{1}{2}\mathbf{c}_{0v}t^2 - \frac{1}{2}\mathbf{c}_{0v}t & (1) \\ \boldsymbol{\omega} = \frac{1}{2}\mathbf{c}_{0\omega}t^2 - \frac{1}{2}\mathbf{c}_{0\omega}t & (2) \end{cases} \quad (\text{B.17})$$

En intégrant (B.17.1) et en utilisant les conditions aux limites suivantes : $\mathbf{t}(0) = \mathbf{t}_0$, $\mathbf{t}(1) = \mathbf{0}$, nous obtenons :

$$\mathbf{t}(t) = (1 + 2t^3 - 3t^2)\mathbf{t}_0 \quad (\text{B.18})$$

En considérant les conditions aux limites, $\mathbf{R}(0) = \mathbf{R}_0$, $\mathbf{R}(1) = \mathbf{I}$ et en introduisant (B.17.2) dans (5.5), et finalement en intégrant (5.5), nous obtenons la trajectoire suivante pour la matrice de rotation :

$$\mathbf{R}(t) = \mathbf{R}_0 \mathbf{e}^{[\mathbf{r}_0]_{\times}(-2t^3 + 3t^2)} \quad (\text{B.19})$$

avec $[\mathbf{r}_0]_{\times} = \log(\mathbf{R}_0^T)$.

Annexe C

Les auto-occultations

Dans cette annexe, nous traitons le problème des auto-occultations en utilisant la même méthode que celle présentée dans le chapitre 6. Dans la première partie, nous définissons la fonction de potentiel répulsif que nous utilisons afin d'éviter les auto-occultations. La seconde section est consacrée à la présentation des résultats expérimentaux obtenus sur la plate-forme robotique de l'IRISA.

C.1 Potentiel répulsif

L'objectif est d'éviter les occultations d'une partie des objets d'intérêt par d'autres objets statiques de la scène lorsque la caméra se déplace. Considérons la projection dans l'image de N éléments \mathcal{P}_j ($j \in \{1 \dots N\}$) de la scène observée par la caméra, nous dirons qu'une occultation apparaît si :

$$\|\mathcal{P}_j - \mathcal{P}_i\| \leq \zeta \quad (\text{C.1})$$

où $i \neq j$ et ζ est une constante. Une classe importante d'occultations est caractérisée par $\mathcal{P}_j \subset \mathcal{T}$ et $\mathcal{P}_i \subset \mathcal{T}$. Elles sont appelées auto-occultations (voir Figure C.1). Les auto-occultations sont détectées en utilisant l'équation (C.1). On appelle \mathcal{O} l'ensemble des configurations telles que :

$$\|\mathcal{P}_j - \mathcal{P}_i\| \leq \zeta + l$$

où l est un réel positif. Afin d'éviter les auto-occultations, le potentiel répulsif V_s est défini par (voir Figure C.2) :

$$\left\{ \begin{array}{ll} -v_s^2 \log \prod_{\substack{i,j=1 \\ j \neq i}}^n \left(1 - \frac{\zeta}{\|\mathcal{P}_j - \mathcal{P}_i\|} \right) & \text{si } \mathbf{s} \notin \mathcal{O} \\ 0 & \text{si } \mathbf{s} \in \mathcal{O} \end{array} \right. \quad (\text{C.2})$$

Comme pour les fonctions de potentiel répulsif définies dans le chapitre 6, v_s est choisie bornée et nulle à la frontière de \mathcal{O} . La fonction V_s est positive ou nulle, tend vers l'infini quand \mathcal{P}_j est proche de \mathcal{P}_i , et est nulle si l'ensemble des primitives visuelles considérées appartient à \mathcal{O} . La force artificielle associée peut aisément être obtenue à partir de (C.2) et de (6.13).

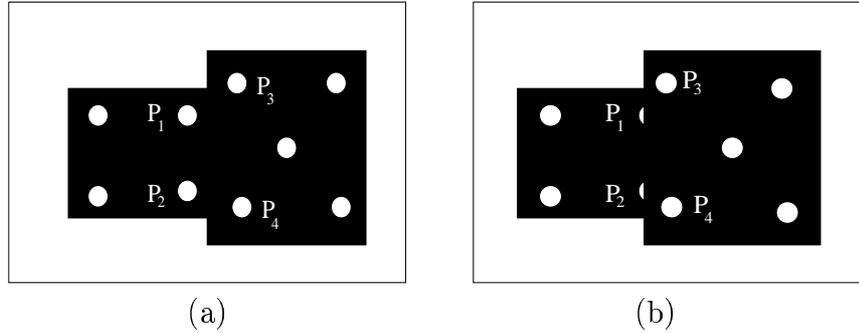


FIG. C.1 – Image de la cible : (a) toutes les primitives visuelles sont visibles, (b) après un mouvement de caméra une partie des primitives visuelles est occultée

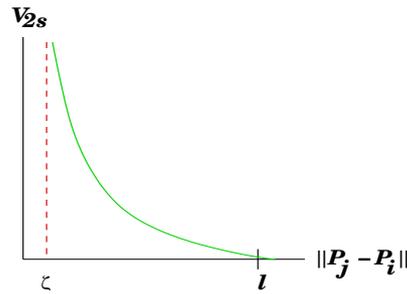


FIG. C.2 – Potentiel répulsif associé à l'évitement des auto-occultations

C.2 Résultats expérimentaux

Nous avons utilisé, pour les expérimentations qui suivent, un objet non plan composé de neuf disques blancs sur fond noir. Les informations visuelles utilisées sont les coordonnées du centre de gravité de chaque disque. La tâche consiste en une tâche de positionnement par rapport à l'objet inconnu non plan. Nous cherchons à obtenir, lors de la phase de planification, des trajectoires dans l'image permettant d'éviter que l'objet d'intérêt sorte du champ de vision de la caméra et les auto-occultations. Les images initiale et désirée de la scène sont illustrées sur les figures C.3(a) et C.3(b), respectivement.

Afin de mettre en évidence l'importance des contraintes introduites lors de la phase

de planification, nous avons, dans un premier temps, réalisé la phase de planification de trajectoires en désactivant les potentiels répulsifs. Nous pouvons observer que les primitives visuelles sortent largement de l'image (se référer à la figure C.4). De telles trajectoires ne sont pas souhaitables puisqu'elles ne permettent pas la réalisation de la tâche.

Nous avons ensuite réalisé la même expérience en activant uniquement le potentiel répulsif associé à la contrainte de visibilité. Dans ce cas, l'objet d'intérêt reste dans l'image (voir figure C.5) mais le point numéro 7 est occulté par le plan 1 (voir figures C.6(a) et C.6(b)).

Dans l'expérimentation dont les résultats sont donnés par la figure C.7, les deux potentiels répulsifs sont activés et nous choisissons $\zeta = d_{min} = 37pixels$ (avec $j = 7$ et $i = 3$, se référer à la relation (C.1)) afin d'éviter l'occultation du point 7 par le plan 1 :

$$\|\mathcal{P}_7 - \mathcal{P}_3\| \leq \zeta$$

L'objet d'intérêt reste alors dans l'image (voir figures C.7(a) et C.7(b)) et le point 7 n'est pas occulté par le plan 1. La distance entre le point 7 et le plan 1 reste toujours supérieure à d_{min} . La phase d'asservissement peut alors être réalisée. Les erreurs sur les coordonnées des points dans l'image sont représentées sur la figure C.7(d). La convergence des coordonnées dans l'image de chacun des points considérés vers leur valeur désirée, démontre que la tâche a été correctement réalisée. L'erreur de suivi est représentée par la figure C.7(e). On remarque que cette erreur reste suffisamment faible (l'erreur maximale reste inférieure à 5 pixels) pour assurer le bon comportement de la loi de commande 2D. Les vitesses de la caméra sont illustrées sur la figure C.7(f).

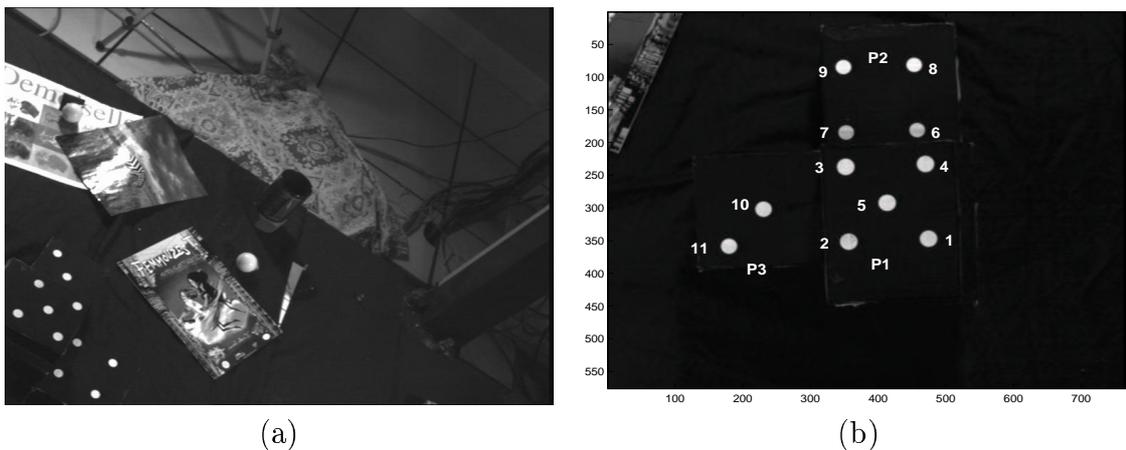


FIG. C.3 – (a) Image initiale et (b) image désirée

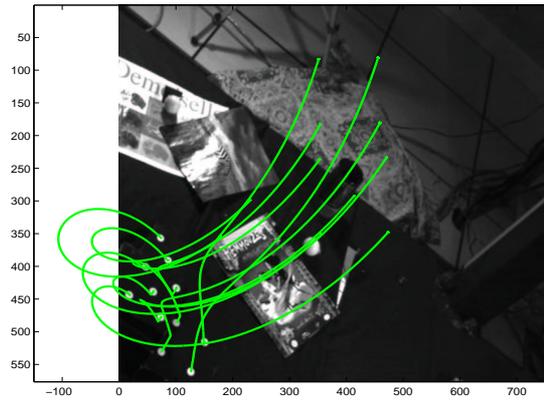


FIG. C.4 – Trajectoires planifiées sans aucun potentiel répulsif

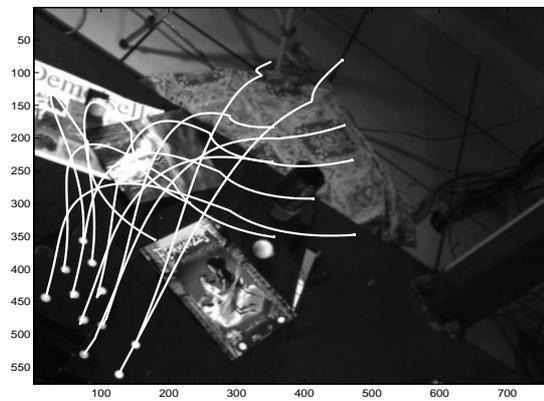
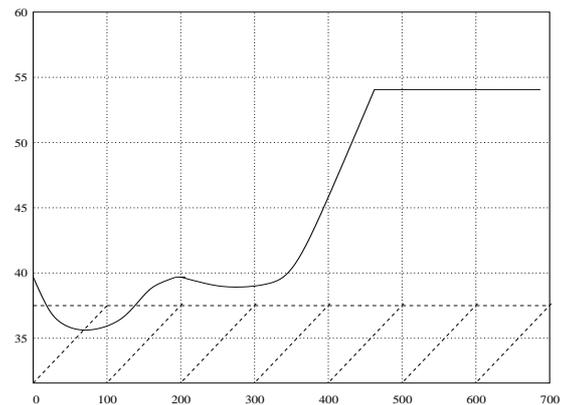


FIG. C.5 – Expérience sans potentiel répulsif associé à l'évitement des occultations : trajectoires planifiées dans l'image



(a)



(b)

FIG. C.6 – Expérience sans potentiel répulsif associé à l'évitement des occultations : (a) le point 7 est occulté par le plan 1, (b) distance entre le point 7 et le point 3

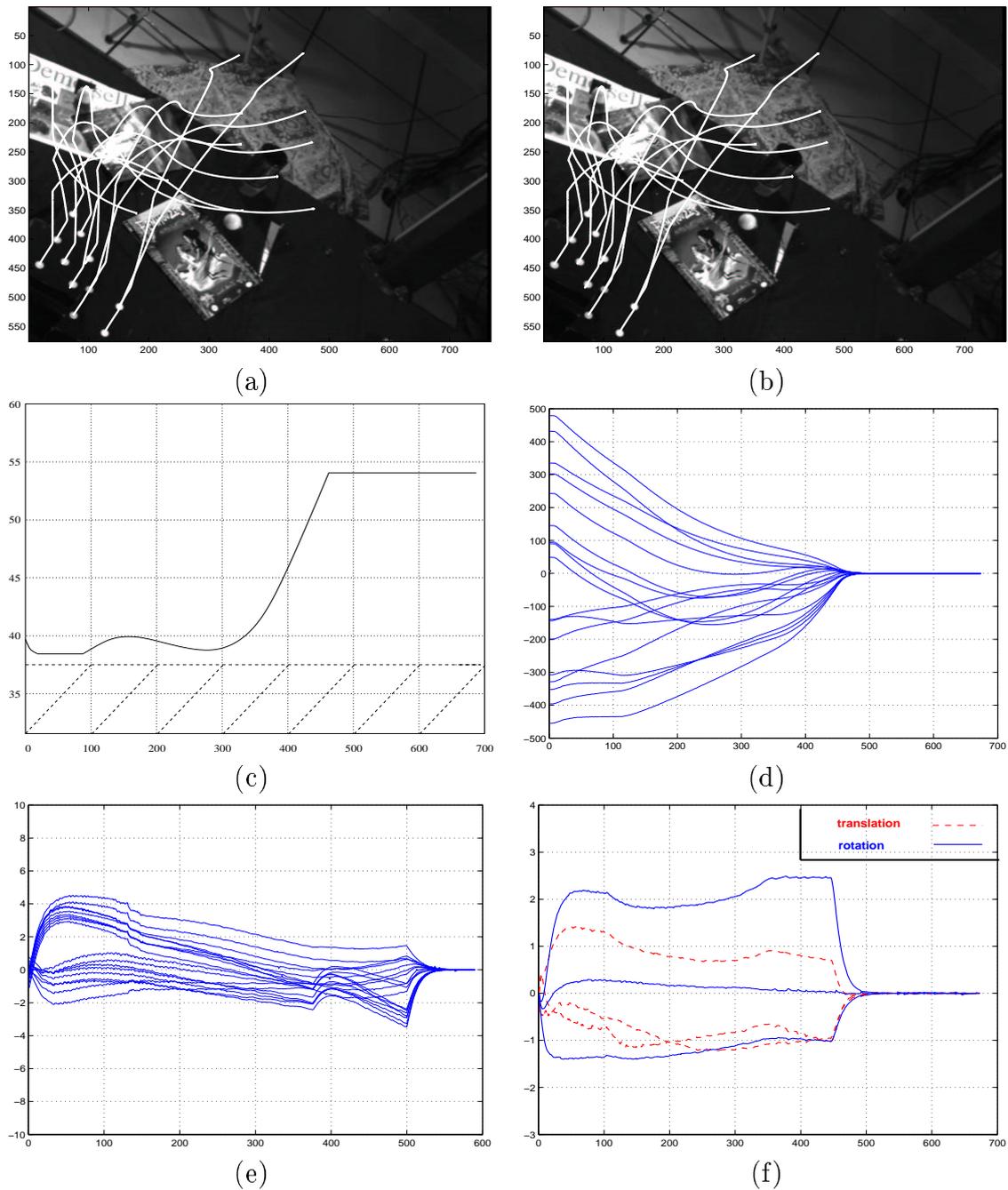


FIG. C.7 – *Expérience avec les potentiels répulsifs associés à l'évitement des occultations et à la contrainte de visibilité: (a) trajectoires planifiées dans l'image, (b) trajectoires suivies dans l'image (c) distance entre le point 7 et le point 3, (d) erreur sur les coordonnées des points, (e) erreurs de suivi (f) vitesses de la caméra (cm/s et deg/s)*

Bibliographie

- [Andreff 00] N. Andreff, B. Espiau, R. Horaud. – Visual servoing from lines. – *IEEE International Conference on Robotics and Automation*, pp. 2070–2075, San Francisco, Californie, avril 2000.
- [Andreff 01] N. Andreff, R. Horaud, B. Espiau. – Robot hand-eye calibration using structure from motion. *International Journal of Robotics Research*, 20(3):228–248, mars 2001.
- [Athans 66] M. Athans, P. L. Falb. – *Optimal control: An introduction to the theory and its applications*. – Lincoln Laboratory Publication, McGraw-Hill Book Compagny, New York, 1966.
- [Avnaim 88] F. Avnaim, J. D. Boissonnat, B. Faverjon. – *A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles*. – Rapport de Recherche INRIA n° 890, Sophia-Antipolis, France, 1988.
- [Barraquand 91] J. Barraquand, J.-C. Latombe. – Robot motion planning: A distributed representation approach. *International Journal Robotics Research*, 10(6):628–649, Décembre 1991.
- [Barraquand 92] J. Barraquand, B. Langlois, J.C. Latombe. – Numerical potential field techniques for robot path planning. *IEEE Transactions on System, Man and Cybernetics*, 22(2):224–241, avril 1992.
- [Basri 98] R. Basri, E. Rivlin, I. Shimshoni. – Visual homing: Surfing on the epipoles. – *IEEE International Conference on Computer Vision*, pp. 863–869, Bombay, Inde, janvier 1998.
- [Bellman 57] R. Bellman. – *Dynamic Programming*. – Princeton University Press, Princeton, New Jersey, 1957.
- [Berkovitz 61] L. D. Berkovitz. – Variational methods in problems of control and programming. *Journal of Mathematical Analysis and Applications*, 3:145–169, 1961.
- [Berry 99] F. Berry. – Contournement d’objet par asservissement visuel. *Thèse de doctorat, Université Blaise Pascal, Clermont II, LASMEA*, février 1999.

- [Berry 00] F. Berry, P. Martinet, J. Gallice. – Real time visual servoing around a complex object. *EICE Transactions on Information and Systems, Special issue on Machine Vision Applications, IEICE'2000*, E83(7):1358–1368, juillet 2000.
- [Bessonnet 94] G. Bessonnet, J.P. Lallemand. – On the optimization of robotic manipulator trajectories with bounded joint actuators or joint kinetic loads considered as control variables. *ASME Journal of Dynamic Systems Measurement and Control*, 1994.
- [Bien 93] Z. Bien, W. Jang, J. Park. – Characterisation and use of feature-jacobian matrix for visual servoing. *Visual Servoing*, éd. par K. Hashimoto, pp. 317–363. – Singapour, World Scientific, 1993.
- [Bishop 94] B. Bishop, S. Hutchinson, M. Spong. – On the performance of state estimation for visual servo systems. – *IEEE International Conference on Robotics and Automation*, pp. 168–173, Los Alamitos, CA, USA, mai 1994.
- [Bliss 61] G. A. Bliss. – *Lectures on the calculus of variations*. – University of Chicago Press, 1st Phoenix édition, Chicago, 1961.
- [Bobrow 85] J.E. Bobrow, S. Dubowsky, J.S. Gibson. – Time-optimal control of robotic manipulators along specified paths. *IEEE International Journal of Robotics Research*, 4(3):3–17, 1985.
- [Borgstadt 01] J. A. Borgstadt, N. J. Ferrier. – Visual servoing by homography decomposition. – *IEEE International Conference on Robotics and Automation*, pp. 723–730, Séoul, Corée du Sud, mai 2001.
- [Boufama 94] B. Boufama. – Reconstruction tridimensionnelle en vision par ordinateur : cas des caméras non étalonnées. *Thèse de doctorat, Institut Polytechnique de Grenoble*, décembre 1994.
- [Boufama 95] B. Boufama, R. Mohr. – Epipole and fundamental matrix estimation using the virtual parallax property. – *IEEE International Conference on Computer Vision*, pp. 1030–1038, Cambridge, USA, 1995.
- [Brooks 83] R.A. Brooks. – Solving the find-path problem by good representation of free space. *IEEE Transactions on Systems, Man and Cybernetics*, 13(3):190–197, 1983.
- [Bryson 63] A. E. Bryson, W. F. Denham, S. E. Dreyfus. – Optimal programming problems with inequality constraints i: Necessary conditions for extremal solutions. *Journal of the American Institute of Aeronautics and Astronautics*, 1(11):2544–2550, 1963.
- [Buckley 85] C. E. Buckley. – The application of continuum methods to

- path planning. *Thèse de doctorat, Université de Stanford*, août 1985.
- [Cadenat 99] V. Cadenat. – Commande référencée multi-capteurs pour la navigation d'un robot mobile. *Thèse de doctorat Université Paul Sabatier de Toulouse, No 3565*, Décembre 1999.
- [Camillo 00] J. T. Camillo, J. P. Ostrowski. – Robust vision-based pose control. – *IEEE International Conference on Robotics and Automation*, vol. 3, San Francisco, Californie, avril 2000.
- [Canny 88] J. F. Canny. – The complexity of robot motion planning. *Thèse de doctorat, MIT Cambridge*, 1988.
- [Cervera 99] E. Cervera, P. Martinet. – Combining pixel and depth information in image-based visual servoing. – *Proceedings of the 9th International Conference on Advanced Robotics, ICAR'99*, pp. 445–450, Tokyo, Japon, Octobre 1999.
- [Chaumette 90] F. Chaumette. – La relation vision-commande: théorie et application à des tâches robotiques. *Thèse de doctorat, Université de Rennes 1, IRISA*, 1990.
- [Chaumette 98] F. Chaumette. – Potential problems of stability and convergence in image-based and position-based visual servoing. *The Confluence of Vision and Control, D. Kriegman, G. Hager, A. Morse (eds), LNCIS Series, Springer Verlag*, 237:66–78, 1998.
- [Chaumette 00a] F. Chaumette, E. Malis. – 2 1/2 d visual servoing: a possible solution to improve image-based and position-based visual servoings. – *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 630–635, San Francisco, Californie, avril 2000.
- [Chaumette 00b] F. Chaumette, E. Marchand. – A new redundancy-based iterative scheme for avoiding joint limits: Application to visual servoing. – *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1720–1725, San Francisco, Californie, avril 2000.
- [Chesi 00] G. Chesi, E. Malis, R. Cipolla. – Automatic segmentation and matching of planar contours for visual servoing. – *International Conference on Robotics and Automation*, pp. 2753–2758, San Francisco, Californie, avril 2000.
- [Chuang 97] D-M. Chuang, S-C. Wu, M-K. Hor. – Adaptive fuzzy visual servoing in robot control. – *IEEE International Conference on Robotics and Automation*, pp. 811–816, Albuquerque, Nouveau Mexique, avril 1997.
- [Collewet 99] C. Collewet. – Contribution à l'élargissement du champ ap-

- plicatif des asservissements visuels 2d. *Thèse de doctorat, Université de Rennes 1, CEMAGREF*, 1999.
- [Colombo 99] C. Colombo, B. Allota. – Image-based robot task planning and control using a compact visual representation. *IEEE Transactions on Systems, Man and Cybernetics*, 29(1):92–100, janvier 1999.
- [Corke 93] Peter Corke. – Visual control of robot manipulators - a review. *Robotics and Automated Systems*. – World Scientific, 1993.
- [Corke 96] P.I. Corke, M.C. Good. – Dynamic effects in visual closed-loop systems. *IEEE Transactions on Robotics and Automation*, 12(5):671–683, octobre 1996.
- [Corke 00] P.I. Corke, S. Hutchinson. – A new partitioned approach to image-based visual servo control. – *IEEE International Conference on Decision and Control*, pp. 2521–2526, Sydney, décembre 2000.
- [Couapel 95] B. Couapel, K. Bainian. – Stereo vision with the use of virtual plane in the space. *Chinese Journal of Electronics*, 4(2):32–39, avril 1995.
- [Cowan 99] N.J. Cowan, D.E. Koditschek. – Planar image based visual servoing as a navigation problem. – *IEEE International Conference on Robotics and Automation*, pp. 611–617, Detroit, Michigan, mai 1999.
- [Crétual 98a] A. Crétual, F. Chaumette. – Image-based visual servoing by integration of dynamic measurements. – *International Conference on Robotics and Automation*, pp. 1994–2001, Louvain, Belgique, mai 1998.
- [Crétual 98b] Armel Crétual. – Asservissement visuel à partir d'informations de mouvement dans l'image. *Thèse de doctorat, Université de Rennes 1, IRISA*, novembre 1998.
- [Crétual 00] Armel Crétual, François Chaumette. – Dynamic stabilization of a pan and tilt camera for submarine image visualization. *Computer Vision and Image Understanding: CVIU*, 79(1):47–65, juillet 2000.
- [Criminisi 98] A. Criminisi, I. Reid, A. Zisserman. – Duality, rigidity and planar parallax. – *European Conference on Computer Vision, ECCV'98*, vol. 2, pp. 846–861, Fribourg, Allemagne, juin 1998.
- [Daucher 97] N. Daucher, M. Dhome, J. T. Lapresté, G. Rives. – Speed command of a robotic system by monocular pose estimate. – *IEEE/RSJ International Conference on Intelligent Robots*

- and Systems*, vol. 1, pp. 55–62, Grenoble, France, septembre 1997.
- [Dementhon 95] D. Dementhon, L.S. Davis. – Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1/2):123–141, juin 1995.
- [Deriche 94] R. Deriche, Z. Zhang, Q.-T. Luong, O. Faugeras. – *Robust recovery of the epipolar geometry for an uncalibrated stereo rig.* – European Conference on Computer Vision, Stockholm, Suède, 1994.
- [Desai 99] J. P. Desai, V. Kumar. – Motion planning for cooperating mobile manipulators. *Journal of Robotic Systems*, 19:557–579, octobre 1999.
- [Dhome 89] M. Dhome, M. Richetin, J.T. Lapresté, G. Rives. – Determination of the attitude of 3d objects from a single perspective image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, décembre 1989.
- [Domingo 96] J. Domingo, J. Pelechano. – Measurement and storage of a network of jacobians as a method for the visual positioning of a robot arm. *Journal of Intelligent and Robotic Systems*, 16:407–422, 1996.
- [Drummond 99] T. Drummond, R. Cippola. – Visual tracking and control using Lie algebra. – *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'99*, vol. 2, pp. 652–657, Fort Collins, Colorado, juin 1999.
- [Espiau 92] B. Espiau, F. Chaumette, P. Rives. – A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, juin 1992.
- [Espiau 95] B. Espiau. – *Sur les erreurs en asservissement visuel.* – Rapport de Recherche INRIA n° 2619, juillet 1995.
- [Fagerer 94] C. Fagerer, D. Dickmanns, E. Dickmanns. – *Visual grasping with long delay time of a free floating object in orbit.* – Autonomous Robots, 1994.
- [Faugeras 86] O. Faugeras, G. Toscani. – The calibration problem for stereo. – *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 15–20, juin 1986.
- [Faugeras 87] O. D. Faugeras, F. Lustman, G. Toscani. – Motion and structure from point and line matches. – *1st International Conference on Computer Vision*, pp. 25–34, Londres, Angleterre, juin 1987.
- [Faugeras 88] O. Faugeras, F. Lustman. – Motion and structure from motion in a piecewise planar environment. *International Journal*

- of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.
- [Faugeras 93] O. Faugeras. – *Three-dimensionnal computer vision: a geometric view point*. – MIT press, Cambridge, Massachusetts, 1993.
- [Feddema 89] J. T. Feddema, O. R. Mitchell. – Vision-guided servoing with feature-based trajectory generation. *IEEE Journal of Robotics and Automation*, 5(5):691–700, octobre 1989.
- [Feddema 90] J. T. Feddema, C. S. G. Lee. – Adaptative image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(5):1172–1183, octobre 1990.
- [Flandin 00] G. Flandin, F. Chaumette, E. Marchand. – Eye-in-hand / eye-to-hand cooperation for visual servoing. – *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2741–2746, San Francisco, Californie, avril 2000.
- [Gangloff 99] J. Gangloff. – Asservissements visuels rapides d’un robot manipulateur à 6 degrés de liberté. *Thèse de doctorat, Université Louis Pasteur, Strasbourg*, Janvier 1999.
- [Gangloff 00] J. Gangloff, M. de Mathelin. – High speed visual servoing of a 6 dof manipulator using multivariable predictive control. – *IEEE International Conference on Robotics and Automation*, pp. 3751–3756, San Francisco, Californie, avril 2000.
- [Gregory 92] J. Gregory, C. Lin. – *Constrained optimization in the calculus of variations and optimal control theory*. – Van Nostrand Reinhold, New York, 1992.
- [Hager 94] G. D. Hager. – Real-time feature tracking and projective invariance as a basis for hand-eye coordination. – *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 533–539, Los Alamitos, Californie, juin 1994.
- [Hager 97] G.D. Hager. – A modular system for robust positioning using feedback from stereo vision. *IEEE Transactions on Robotics and Automation*, 13(4):582–595, août 1997.
- [Harris 88] C. Harris, M. Stephens. – A combined corner and edge detector. – *Alvey Conference*, pp. 189–192, 1988.
- [Hartley 92] R. I. Hartley. – *Estimation of relative camera positions for uncalibred cameras*. – G. Sandini (édité par), Computer Vision - ECCV’92, vol. 588 of Lecture Notes in Computer Science, 579–587p., Santa Margherita Ligure, Italie, mai 1992.
- [Hartley 97] R. I. Hartley. – In defense of the height-point algorithm.

- IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, juin 1997.
- [Hashimoto 91] K. Hashimoto, T. Kimoto, T. Ebine, H. Kimura. – Manipulator control with image-based visual servo. – *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2267–2271, Sacramento, Californie, avril 1991.
- [Hashimoto 93a] K. Hashimoto. – *Visual Servoing: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. – World Scientific Series in Robotics and Automated Systems, Vol. 7, World Scientific Press, Singapour, 1993.
- [Hashimoto 93b] K. Hashimoto, H. Kimura. – Dynamic visual servoing with nonlinear model-based control. – *Proceedings of the 12th World Congress IFAC*, vol. 9, pp. 405–408, Sydney, Australie, juillet 1993.
- [Hashimoto 93c] K. Hashimoto, H. Kimura. – LQ optimal and nonlinear approaches to visual servoing. *Visual Servoing*, éd. par K. Hashimoto, pp. 165–198. – World Scientific Press, 1993.
- [Hashimoto 96] K. Hashimoto, T. Ebine, H. Kimura. – Visual servoing with hand-eye manipulator - optimal control approach. *IEEE Transactions on Robotics and Automation*, 12(5):766–774, octobre 1996.
- [Hashimoto 97] K. Hashimoto, T. Noritugu. – Visual servoing of nonholonomic cart. – *IEEE International Conference on Robotics and Automation*, pp. 1719–1724, Albuquerque, Nouveau Mexique, avril 1997.
- [Hashimoto 98] K. Hashimoto, T. Noritsugu. – Performance and sensitivity in visual servoing. – *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2321–2326, Louvain, Belgique, mai 1998.
- [Horaud 93] R. Horaud, O. Monga. – *Vision par ordinateur, outils fondamentaux*. – Hermès, Paris, France, 1993.
- [Horaud 98] R. Horaud, F. Dornaika, B. Espiau. – Visually guided object grasping. *IEEE Transactions on Robotics and Automation*, 14(4):525–532, août 1998.
- [Hosoda 94] K. Hosoda, M. Asada. – Versatile visual servoing without knowledge of true jacobian. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'94*, pp. 186–193, septembre 1994.
- [Hosoda 95] K. Hosoda, K. Sakamoto, M. Asada. – Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction. *IEEE/RSJ International Confe-*

- rence on Intelligent Robots and Systems*, 1(3):29–34, août 1995.
- [Huang 89] T. S. Huang, O. Faugeras. – Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, décembre 1989.
- [Hutchinson 96] S. Hutchinson, G.D. Hager, P.I. Corke. – A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, octobre 1996.
- [Jacobson 69] D. H. Jacobson, M. M. Lele. – A transformation technique for optimal control problems with state variable inequality constraint. *IEEE Transactions on Automatic Control*, 14(5):457–464, 1969.
- [Jägersand 97] M. Jägersand, O. Fuentes, R. Nelson. – Experimental evaluation of uncalibrated visual servoing for precision manipulation. – *IEEE International Conference on Robotics and Automation*, pp. 2874–2880, Albuquerque, Nouveau Mexique, avril 1997.
- [Jerian 91] C.P. Jerian, R. Jain. – Structure from motion: A critical analysis of methods. *SMC*, 21:572–588, 1991.
- [Kahn 71] M.E. Kahn, B. Roth. – The near-minimum-time control of open-loop articulated kinematic chains. *ASME Journal of Dynamic Systems, Measurement, and Control*, 93:164–172, septembre 1971.
- [Kavraki 96] L. Kavraki, M.N. Kolountzakis, J.-C. Latombe. – Analysis of probabilistic roadmaps for path planning. – *IEEE International Conference on Robotics and Automation*, pp. 3020–3026, Minneapolis, MN, 1996.
- [Kelly 96] R. Kelly. – Robust asymptotically stable visual servoing of planar robots. *IEEE Transactions on Robotics and Automation*, 12(5):755–766, octobre 1996.
- [Khatib 80] O. Khatib. – Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles. *Thèse de doctorat, Ecole Nationale Supérieure de l'Aréonautique et de l'Espace, Toulouse*, 1980.
- [Khatib 86] O. Khatib. – Real time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.
- [Koditschek 87] D. E. Koditschek. – Exact robot navigation by means of potential functions: Some topological considerations. – *International Conference on Robotics and Automation*, pp. 1–6, Raleigh, NC, 1987.

- [Koivo 91] A. J. Koivo, N. Houshangi. – Real-time vision feedback for servoing robotic manipulator with self-tuning controller. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(1):134–142, février 1991.
- [Langlois 92] B. Langlois, J. Barraquand, J.-C. Latombe. – Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):224–241, 1992.
- [Latombe 91] J. C. Latombe. – *Robot Motion Planning*. – Kluwer Academic Publishers, 1991.
- [Laumond 87] J.P. Laumond. – Obstacle growing in a non-polygonal world. *Information Processing Letters*, 25(1):41–50, 1987.
- [Laumond 94] J-P. Laumond, P. Jacobs, M. Taïx, P. Murray. – A motion planner for nonholonomic mobile robots. *IEEE Transaction on Robotics and Automation*, 10, 1994.
- [Li 95] M. Li, J.-M. Lavest. – *Some aspects of zoom-lens camera calibration*. – Rapport de Recherche n° ISRN KTH/NA/P-95/03-SE, Computational Vision and Active Perception Laboratory, Royal Institute of Technology, Stockholm, Sweden, février 1995.
- [LonguetHiggins 81] H.C. Longuet-Higgins. – A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, septembre 1981.
- [LonguetHiggins 84] H. C. Longuet-Higgins. – The reconstruction of a scene from two projections: configurations that defeat the 8-point algorithm. – *1st Conference on Artificial Intelligence Applications*, pp. 395–397, Denver, 1984.
- [Lowe 91] D. G. Lowe. – Fitting parameterized three-dimensional models to image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, mai 1991.
- [LozanoPerez 79] T. Lozano-Perez, M. A. Wesley. – An algorithm for planning collision-free paths among polyhedral obstacles. *Communication of the ACM*, 22(10):560–570, 1979.
- [Luong 93] Q-T. Luong, R. Deriche, O. Faugeras, T. Papadopoulos. – On determining the fundamental matrix: analysis of different methods and experimental results. – *Israeli Conference on Artificial Intelligence and Computer Vision*, Tel-Aviv, Israël, 1993.
- [Luong 96] Q-T. Luong, O. Faugeras. – The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1), janvier 1996.

- [Malis 98] E. Malis. – Contributions à la modélisation et à la commande en asservissement visuel. *Thèse de doctorat, Université de Rennes 1, IRISA*, novembre 1998.
- [Malis 99] E. Malis, F. Chaumette, S. Boudet. – 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, avril 1999.
- [Malis 00] E. Malis, F. Chaumette. – 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97, juin 2000.
- [Marchand 96] E. Marchand, F. Chaumette, A. Rizzo. – Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'96*, pp. 1083–1090, Osaka, Japon, novembre 1996.
- [Marchand 98] E. Marchand, G.D. Hager. – Dynamic sensor planning in visual servoing. – *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1988–1993, Louvain, Belgique, mai 1998.
- [Marchand 01] E. Marchand, F. Chaumette, F. Spindler, M. Perrier. – Controlling the manipulator of an underwater ROV using a coarse calibrated pan-tilt camera. – *International Conference on Robotics and Automation*, pp. 2773–2778, Séoul, Corée du Sud, mai 2001.
- [Martin 89] D. P. Martin, J. Baillieul, J. M. Hollerbach. – Resolution of kinematic redundancy using optimization techniques. *IEEE Transactions on Robotics and Automation*, 5(4):529–533, août 1989.
- [Martinet 96] P. Martinet, J. Gallice, D. Khadraoui. – Vision-based control. – *Proceedings of the World Automation Congress, WAC'96, Robotic and Manufacturing Systems*, vol. 3, pp. 497–502, Montpellier, France, mai 1996.
- [Martinet 97] P. Martinet, N. Daucher, J. Gallice, M. Dhome. – Robot control using monocular pose estimation. – *Workshop on New Trends in Image-based Robot Servoing, IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'97*, pp. 1–12, Grenoble, France, Septembre 1997.
- [Martinet 99] P. Martinet, J. Gallice. – Position-based visual servoing using a nonlinear approach. – *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 531–536, Kyongju, Corée du Sud, octobre 1999.
- [Mezouar 02] Y. Mezouar, A. Remazeilles, P. Gros, F. Chaumette. – Image

- interpolation for image-based control under large displacement. *IEEE International Conference on Robotics and Automation*, mai 2002.
- [Michel 93] H. Michel, P. Rives. – *Singularities in the determination of the situation of a robot effector from the perspective view of 3 points*. – Rapport de Recherche INRIA n° 1850, février 1993.
- [Morel 00] G. Morel, T. Leibzeit, J. Szewczyk, S. Boudet, J. Pot. – Explicit incorporation of 2d constraints in vision-based control of robot manipulators. – P. Corke, J. Trevelyan (édité par), *Experimental Robotics VI*, pp. 99–108. Springer Verlag, 2000.
- [Nakamura 87] Y. Nakamura, H. Hanafusa. – Optimal redundancy control of robot manipulators. *International Journal of Robotics Research*, 6, 1987.
- [Nelson 96] B.J. Nelson, P.K. Khosla. – Vision resolvability for visually servoed manipulation. *Journal of Robotic System*, 13(2):73–93, 1996.
- [Overmars 94] M.H. Overmars, P. Svestka. – A probabilistic learning approach to motion planning. – A. K. Peters (édité par), *The First Workshop on the Algorithmic Foundations of Robotics*, pp. 19–37, Boston, MA, 1994.
- [Papanikolopoulos 91] N. Papanikolopoulos, P.K. Khosla, T. Kanade. – Vision and control techniques for robotic visual tracking. – *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 857–864, Sacramento, Californie, avril 1991.
- [Papanikolopoulos 93] N. P. Papanikolopoulos, P. K. Khosla. – Adaptive robotic visual tracking: Theory and experiments. *IEEE Transactions on Automatic Control*, 38(3):429–445, mars 1993.
- [Papanikolopoulos 94] N. P. Papanikolopoulos, B. Nelson, P. K. Khosla. – Six-degree-of-freedom hand/eye visual tracking with uncertain parameters. – *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 174–180, Los Alamitos, Californie, mai 1994.
- [Pontryagin 62] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishchenko. – *The mathematical theory of optimal processes*. – Interscience Publishers, New York, 1962.
- [Rimon 92] E. Rimon, D. E. Koditschek. – Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.
- [Rives 97] P. Rives, J-J. Borrelly. – Visual servoing techniques applied to an underwater vehicle. – *IEEE International Conference on Robotics and Automation*, pp. 1851–1856, Albuquerque, Nouveau Mexique, USA, avril 1997.

- [Rizzi 96] A. Rizzi, D. Koditschek. – An active visual estimator for dexterous manipulation. *IEEE Transactions on Robotics and Automation*, 12(5):697–713, 1996.
- [Rémy 98] S. Rémy. – Etalonnage d'un système de vision embarqué. *Thèse de doctorat, Université Blaise Pascal, Clermont II*, juillet 1998.
- [Robert 95] L. Robert, C. Zeller, O.D. Faugeras, M. Hebert. – *Applications of Non-Metric Vision to Some Visually-Guided Robotics Tasks*. – Rapport de Recherche INRIA n° 2584, 1995.
- [Rosen 76] C. Rosen. – *Machine Intelligence research applied to industrial automation*. – Rapport de recherche, SRI International, Sixth report, 1976.
- [Ruf 97] A. Ruf, R. Horaud. – Visual trajectories from uncalibrated stereo. – *Workshop on New Trends in Image-based Robot Servoing, IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'97*, vol. 2, pp. 83–91, Grenoble, France, septembre 1997.
- [Ruf 00] A. Ruf, R. Horaud. – Vision-based guidance and control of robots in projective space. – Springer (édité par), *6th European Conference on Computer Vision (ECCV'00)*, vol. 2, pp. 50–66, Dublin, Irlande, juin 2000.
- [Sagan 69] H. Sagan. – *Introduction to the calculus of variations*. – McGraw-Hill Book Company, New York, 1969.
- [Samson 91] C. Samson, B. Espiau, M. Le Borgne. – *Robot Control: The Task Function Approach*. – Oxford University Press, 1991.
- [Sanderson 80] A.C Sanderson, L.E Weiss. – Image-based visual servo control using relational graph error signals. – *International Conference on Cybernetics and Society*, pp. 1074–1077, Cambridge, MA, Octobre 1980.
- [Sanderson 83] A.C Sanderson, L.E Weiss. – *Adaptive visual servo control of robots - Robot Vision*. – édité par A. Pugh, Bedford, UK, 1983.
- [SantosVictor 97] J. Santos-Victor, G. Sandini. – Visual behaviors for docking. *Computer Vision and Image Understanding*, 67(3):223–238, 1997.
- [Schmid 97] C. Schmid, R. Mohr. – Local greyvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, mai 1997.
- [Schumaker 81] L. Schumaker. – *Spline functions: Basic Theory*. – Wiley, 1981.
- [Sharma 95] R. Sharma, S. Hutchinson. – Optimizing hand/eye configuration for visual-servo systems. – *IEEE International Confe-*

- rence on Robotics and Automation*, vol. 1, pp. 172–177, Nagoya, Japon, 1995.
- [Shashua 94] A. Shashua, N. Navab. – Relative affine structure: Theory and application to 3d reconstruction from perspective views. – *IEEE International Conference on Vision and Pattern Recognition*, pp. 483–489, 1994.
- [Shiller 91] Z. Shiller, S. Dubowsky. – On computing the global time-optimal control of robotic manipulators with geometric path constraint. *IEEE Transactions on Robotics and Automation*, 7(6):785–797, décembre 1991.
- [Shirai 73] Y. Shirai, H. Inoue. – Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5:99–108, 1973.
- [Singh 89] S. Singh, M. C. Leu. – Optimal trajectory generation for robotic manipulators using dynamic programming. *ASME Journal of Dynamic Systems, Measurement, and Control*, 1989.
- [Singh 98] R. Singh, R. M. Voyle, D. Littau, N. P. Papanikolopoulos. – Alignment of an eye-in-hand system to real objects using virtual images. – *Workshop on Robust Vision for Vision-Based Control of Motion, IEEE International Conference on Robotics and Automation*, Louvain, Belgique, mai 1998.
- [Slotine 89] J.-J. E. Slotine, H. S. Yang. – Improving the efficiency of time-optimal path-following algorithms. *IEEE Transactions on Robotics and Automation*, 5(1):118–124, février 1989.
- [Smith 97] C.E. Smith, S.A. Brandt, N.P. Papanikolopolulos. – Eye-in-hand robotic tasks in uncalibrated environments. *Journal of Intelligent and Robotic Systems*, 19:237–270, 1997.
- [Souères 97] P. Souères, J-D. Boissonnat. – *Optimal trajectories for non-holonomic mobile robots*. – Robot Motion Planning and Control, Springer, No. 229 de Lecture Notes in Control and Information Sciences, pp. 93-170, novembre 1997.
- [Suh 87] K.C. Suh, J. M. Hollerbach. – Local versus global torque optimization of redundant manipulators. – *IEEE International Conference on Robotics an Automation*, Raleigh, NC, 1987.
- [Suh 93] I. H Suh, T. W. Kim. – Visual servoing of robot manipulators by fuzzy membership function based neural networks. *Visual Servoing*, éd. par K. Hashimoto, pp. 165–198. – World Scientific Press, 1993.
- [Sundareswaran 94] V. Sundareswaran, F. Chaumette, P. Bouthemy. – Visual servoing using image motion information. – *IEEE International Workshop on Visual Behaviors*, pp. 102–106, Seattle, Washington, juin 1994.

- [Tani 77] K. Tani, M. Abe, K. Tanie, T. Ohno. – High precision manipulators with visual sense. – *7th International Symposium on Industrial Robotics*, pp. 561–568, Tokyo, Japon, octobre 1977.
- [Tarbouriech 00] S. Tarbouriech, P. Souères. – Advanced control strategy for the visual servoing scheme. – *6th IFAC Symposium on Robot Control, SYROCO'00*, Vienna, Austria, 2000.
- [Torr 97] P. H. S Torr, D. W. Murray. – The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, septembre 1997.
- [Torr 98] P. H. S Torr, A. Zisserman. – Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding*, 71(3):312–333, 1998.
- [Tsai 86] R. Y. Tsai. – An efficient and accurate camera calibration technique for 3d machine vision. – *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 364–374, Miami, juin 1986.
- [Tsakiris 98] D. Tsakiris, P. Rives, C. Samson. – Extending visual servoing techniques to nonholonomic mobile robots. *The Confluence of Vision and Control*, éd. par G. Hager D. Kriegman, A. Morse, pp. 106–117. – Springer Verlag, 1998.
- [Valentine 37] F. A. Valentine. – The problem of Lagrange with differential inequalities as added side conditions. *Contribution to the Calculus of variations*, pp. 407–448. – Chicago University Press, 1937.
- [Viéville 96] T. Viéville, C. Zeller, L. Robert. – Using collineations to compute motion and structure in an uncalibrated image sequence. *International Journal of Computer Vision*, 20(3):213–242, 1996.
- [Vutobratović 82] M. Vutobratović, M. Kirčanski. – A method for optimal synthesis of manipulation robot trajectories. *ASME Journal of Dynamic Systems, Measurement, and Control*, 104, 1982.
- [Walsh 94] G. Walsh, D. Tilbury, S. Sastry, R. Murray, J. P. Laumond. – Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 39(1):216–222, 1994.
- [Weiss 87] L.E. Weiss, A.C. Sanderson, C.P. Neuman. – Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417, octobre 1987.

- [Wells 96] G. Wells, C. Venaille, C. Torras. – Promising research vision-based robot positioning using neural networks. *Image and Vision Computing*, 14:714–732, 1996.
- [Wijesoma 93] S. W Wijesoma, D. F. H Wolfe, R. J Richards. – Eye-to-hand coordination for vision-guided robot control applications. *International Journal of Robotics Research*, 12(1):65–78, février 1993.
- [Wilson 96] W.J Wilson, C.C Williams Hulls, G.S Bell. – Relative end-effector control using Cartesian position-based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, 1996.
- [Wu 97] Q. M. J Wu, K. Stanley. – Modular neural-visual servoing using a neural-fuzzy decision network. – *IEEE International Conference on Robotics and Automation*, pp. 3238–3243, Albuquerque, Nouveau Mexique, avril 1997.
- [Wunsch 97] P. Wunsch, S. Winkler, G. Hirzinger. – Real-time pose estimation of 3-d objects from camera images using neural networks. – *IEEE International Conference on Robotics and Automation*, pp. 3232–3237, Albuquerque, Nouveau Mexique, avril 1997.
- [Yuan 89] J. S. C. Yuan. – A general photogrammetric method for determining object position and orientation. *IEEE Transactions on Robotics and Automation*, 5(2):129–142, avril 1989.
- [Zanne 00] P. Zanne, G. Morel, F. Plestan. – Robust vision based 3d trajectory tracking using sliding mode control. – *IEEE International Conference on Robotics and Automation*, pp. 2088–2093, San Francisco, Californie, avril 2000.
- [Zefran 96] M. Zefran, V. Kumar. – Planning smooth motions on $SE(3)$. – *IEEE International Conference on Robotics and Automation*, pp. 121–126, Minneapolis, avril 1996.
- [Zefran 97] M. Zefran, V. Kumar. – A variational calculus framework for motion planning. – *International Conference on Advanced Robotics*, Monterey, Californie, juillet 1997.
- [Zhang 95a] Z. Zhang, R. Deriche, O. Faugeras, Q.-T. Luong. – A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, octobre 1995.
- [Zhang 95b] Z. Zhang, A. R. Hanson. – Scaled euclidean 3d reconstruction based on externally uncalibrated cameras. *IEEE Symposium on Computer Vision*, Coral Gables, Floride, 1995.
- [Zhang 96a] Z. Zhang. – *Determining the epipolar geometry and its un-*

- certainty: a review.* – Rapport de Recherche INRIA n° 1996, juillet 1996.
- [Zhang 96b] Z. Zhang. – On the epipolar geometry between two images with lens distorsion. – *International Conference on Pattern Recognition*, vol. 1, pp. 407–411, Vienne, Autriche, août 1996.
- [Zhang 96c] Z. Zhang, Q.-T. Luong, O. Faugeras. – Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. *IEEE Transactions on Robotics and Automation*, 12(1):103–113, février 1996.
- [Zhu 91] D. Zhu, J.-C. Latombe. – New heuristic algorithms for efficient hierarchial path planning. *IEEE Transactions on Robotics and Automation*, 7:9–20, février 1991.

Publications

Journaux

1. Y. Mezouar, F. Chaumette. – Path planning for robust image-based Control. – *A paraître dans IEEE Transactions on Robotics and Automation.*
2. Y. Mezouar, F. Chaumette. – Avoiding self-occlusions and preserving visibility by path Planning in the Image. – *A paraître dans Robotics and Autonomous Systems Journal.*

Congrès internationaux

3. Y. Mezouar, A. Remazeilles, P. Gros, F. Chaumette – Images interpolation for image-based control under large displacement. – *IEEE International Conference on Robotics and Automation, ICRA '02*, Volume 3, pages 3787-3794, Washington DC, Mai 2002.
4. Y. Mezouar, F. Chaumette – Model-free optimal trajectories in the image space: application to robot vision control – *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'01*, Volume 1, pages 1155-1162, Hawaii, décembre 2001.
5. Y. Mezouar, F. Chaumette. – Model-free optimal trajectories in the image space. – *IEEE International Conference on Intelligent Robots and Systems, IROS'01*, Volume 1, pages 25-31, Hawaii, octobre 2001.
6. Y. Mezouar, F. Chaumette. – Visual servoing by path planning.– *European Conference on Control, ECC'01*, pages 2904-2909, Porto, Portugal, Septembre 2001.
7. Y. Mezouar, F. Chaumette – avoiding self-occlusions and preserving visibility by path planning in the Image. – *9th International Symposium on Intelligent Robotic systems, SIRS'01*, Toulouse, Juillet 2001.
8. Y. Mezouar, F. Chaumette – Design and tracking of desirable trajectories in the image space by integrating mechanical and visibility constraints. – *IEEE International Conference on Robotics and Automation, ICRA'01*, Volume 1, pages 731-736, Séoul, Corée du Sud, avril 2001.
9. Y. Mezouar, F. Chaumette – Path planning in image space for robust visual servoing. – *IEEE International Conference on Robotics and Automation,*

ICRA '00, Volume 3, pages 2759-2764, San Fransisco, avril 2000.

Rapport de recherche

10. Y. Mezouar, F. Chaumette. – Path planning for robust image-based visual servoing. – *Rapport de recherche IRISA-INRIA*, No 1377, Janvier 2001.

Congrès nationaux

11. Y. Mezouar, F. Chaumette. – Génération de trajectoire dans l'image : vers l'asservissement visuel robuste. – *12ème Journées des Jeunes Chercheurs en Robotique*, pages 131-137, Bourges, février 2000.
12. Y. Mezouar. – Comparaison des lois de commande 3D et 2D en asservissement visuel. – *9ème Journées des Jeunes Chercheurs en Robotique*, Clermont-Ferrand, Juin 1998.

Soumissions

13. Y. Mezouar, F. Chaumette – Optimal camera trajectory with image based control. – *Soumis à International Journal of Robotics Research*.
14. Y. Mezouar, F. Chaumette – Enlarging the stable region of image-based control by path planning. – *Fifteenth International Symposium on Mathematical Theory of Networks and Systems*, Indiana, aout 2002.

Planification de trajectoires pour l'asservissement visuel

Ce travail de thèse s'inscrit dans le domaine de la vision robotique. Plus précisément, nous nous sommes intéressés à améliorer les techniques actuelles d'asservissement visuel. L'asservissement visuel consiste à contrôler un système en utilisant des informations extraites des images fournies par une ou plusieurs caméras. Les techniques classiques sont basées sur la régulation à zéro de l'erreur entre les mesures courante et désirée. Ces méthodes se heurtent à des difficultés en particulier lorsque la configuration à atteindre est très éloignée de la configuration courante. Pour pallier ces difficultés, nous proposons le couplage d'une phase de planification de trajectoires des primitives visuelles dans l'image et d'une phase de suivi de trajectoires par asservissement visuel basé image. Nous proposons, à cette fin, deux schémas de planification de trajectoires. La première méthode repose sur le formalisme de la commande optimale. Le principe du second schéma de planification est basée sur la méthode des fonctions de potentiel. Pour les deux techniques proposées, les trajectoires dans l'image sont peu influencées par la calibration de la caméra, et les modèles des objets observés ne sont pas nécessaires. Nous considérons également les questions d'optimalité et les problèmes liés à la visibilité des objets d'intérêt et à l'évitement des butées articulaires. En exploitant le bon comportement des asservissements 2D aux petites erreurs, nous avons obtenu des schémas de commande robustes pour le suivi de trajectoires. Les méthodes que nous avons proposées ont été testées en utilisant une caméra montée sur un robot cartésien à six degrés de liberté. Les résultats obtenus confirment que le couplage planification-suivi dans l'image permet de réaliser des tâches robotiques à l'aide d'un capteur de vision lorsque les déplacements à effectuer et les erreurs de modélisation sont très importants.

Mots-Clefs : Vision, Robotique, Asservissement visuel, Planification de trajectoires

Path planning for visual servoing

This thesis is concerned with robot vision. More precisely, we are interested in path planning for visual servoing. Visual control consists in controlling a system by using informations extracted from images provided by one or more cameras. Classical approaches are based on the regulation to zero of an error function computed from the current measurement and the desired one. By using such approaches, it is not obvious to introduce any constraint in the realised trajectories and to ensure the convergence whatever the displacement that has to be realised. To resolve these deficiencies of visual servoing techniques, we propose to couple path planning in image space and image-based control. We have proposed two path planning schemes. The first one is based on the optimal control formalism. The second one is based on the potential field method. The proposed schemes are model-free and are robust with respect to errors on internal camera parameters. Optimality issues have been studied and constraints such that the object of interest remains in the camera field of view or the robot avoids its joints limits have been introduced at the task planning level. By exploiting the well known good behaviour of image-based control for small errors, we have obtained robust control schemes for trajectories following in the image space. Real time experimental results obtained on a six DOF eye-in-hand robotic system validate the proposed approach.

Mots-Clefs : Vision, Robotics, Visual servoing, Path planning