

THÈSE

présentée

DEVANT L'UNIVERSITÉ DE RENNES I

pour obtenir

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

Mention : Informatique

par

Omar Tahri

Équipe d'accueil : VISTA (IRISA, RENNES)

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

Composante Universitaire : Institut de Formation Supérieur en Informatique et
Communication

Titre de la thèse :

**Utilisation des moments en asservissement visuel
et en calcul de pose.**

soutenue le 22 mars 2004, devant la commission d'Examen

COMPOSITION DU JURY :

M.	Patrick	BOUTHEMY	Président
M.	Phillipe	MARTINET	Rapporteur
M.	Patrick	RIVES	Rapporteur
M.	Seth	HUTCHINSON	Examineur
M.	Peter	STURM	Examineur
M.	François	CHAUMETTE	Directeur de thèse

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 ici :

Patrick Bouthemy, Directeur de Recherche INRIA à l'IRISA/INRIA de Rennes, pour avoir accepté de présider ce jury de thèse, mais également pour m'avoir accueilli dans son équipe de recherche.

Philippe MARTINET, Professeur au sein de l'Université Blaise Pascal de Clermont-Ferrand pour avoir rapporté cette thèse, pour les discussions constructives que nous avons pu avoir, et pour tout l'intérêt qu'il a accordé à cette thèse,

Patrick RIVES, Directeur de Recherche à l'INRIA Sophia Antipolis, pour avoir accepté d'être rapporteur de cette thèse et pour toutes ces remarques pertinentes sur ce travail,

François Chaumette, Directeur de Recherche INRIA à l'IRISA/INRIA de Rennes, pour m'avoir proposé ce sujet et pour m'avoir encadré avec intérêt le long de cette thèse. Je tiens aussi à le remercier pour ses remarques promptes et pertinentes pendant nos discussions et tout l'effort qu'il a porté à la correction de ce manuscrit,

Seth Hutchinson, Professeur d'Université au sein de l'Université d'Illinois à Urbana-Champaign, USA, pour sa participation au jury de thèse,

Peter Sturm, chargé de recherche à l'INRIA Rhône-Alpes, pour sa présence et sa participation au jury de thèse,

Tous mes collègues qui ont contribué à la relecture de ce manuscrit : Gwen, Vincent, Thomas, Andrew, Nicolas, Tony et Ewa, ainsi que tous les membres de l'équipe Vista avec lesquels j'ai partagé de l'amitié et des moments agréables.

Table des matières

1	Introduction	9
1.1	Introduction générale	9
1.2	Asservissement visuel	10
1.2.1	Asservissement visuel 2 D	10
1.2.2	Asservissements visuels 3 D et 2 D 1/2	12
1.2.3	Travaux récents en asservissement visuel	13
1.3	Calcul de pose	14
1.4	Les moments comme informations visuelles	15
1.5	Organisation du manuscrit	16
2	Les moments invariants	17
2.1	Définitions	17
2.1.1	Moments 2D: définitions et propriétés	17
2.1.2	Moments 3D et moments k-dimensionnels	21
2.2	Calcul des moments invariants	23
2.2.1	Introduction	23
2.2.2	Théorie des invariants algébriques	24
2.2.2.1	Concepts	24
2.2.2.2	Le théorème fondamental des moments invariants révisé	25
2.2.3	Méthode générale de calcul des moments invariants	26
2.2.3.1	Invariants aux transformations orthogonales	26
2.2.3.2	Élimination de la redondance	31
2.2.3.3	Invariance à l'échelle	32
2.2.3.4	Invariance aux rotations et à l'échelle	33
2.3	Généralisation au cas d'un ensemble de points	34
2.3.1	Définitions	34
2.3.1.1	Moments 2D	34
2.3.1.2	Moments k-dimensionnels	35
2.3.2	Théorème fondamental des moments invariants d'un ensemble de points	36
2.3.3	Invariants aux rotations	36
2.3.4	Invariants aux changements d'échelle	37

2.4	Conclusion	38
3	Asservissement visuel à partir des moments	39
3.1	Introduction	39
3.2	Matrice d'interaction et loi de commande	40
3.2.1	Cas où \mathbf{s} est de dimension supérieure à 6 ($k > 6$)	42
3.2.2	Cas où \mathbf{s} est de dimension égale à 6 ($k = 6$)	43
3.2.3	Choix des informations visuelles	44
3.3	Matrice d'interaction de primitives usuelles	45
3.3.1	Matrice d'interaction d'un point	45
3.3.2	Matrice d'interaction d'une primitive géométrique paramétrable : cas d'une droite	47
3.3.3	Matrice d'interaction 3D	50
3.3.3.1	Matrice d'interaction associée à $\theta \mathbf{u}$	50
3.3.3.2	Matrice d'interaction associée à un point 3D	51
3.3.3.3	Informations visuelles 2D 1/2	51
3.4	Matrice d'interaction associée au moment	53
3.4.1	Modélisation de l'objet	53
3.4.2	Matrice d'interaction des moments calculés à partir d'une pro- jection perspective	54
3.4.2.1	Matrice d'interaction des moments 2D: cas continu	54
3.4.2.2	Matrice d'interaction des moments 2D: cas discret	59
3.4.3	Matrice d'interaction des moments calculés à partir d'une pro- jection sphérique	62
3.4.3.1	Expression des moments en projection sphérique	62
3.4.3.2	Matrice d'interaction des moments d'une projection sphérique: cas continu	65
3.4.3.3	Matrice d'interaction des moments d'une projection sphérique: cas discret	66
3.5	Choix des informations visuelles	67
3.5.1	Utilisation des invariants en projection sphérique	67
3.5.2	Informations visuelles pour commander les translations	70
3.5.3	Informations visuelles pour commander les rotations	71
3.6	Choix des informations visuelles dans le cas discret	73
3.7	Cas d'une position désirée de l'objet non parallèle au plan image	75
3.8	Conclusion	79
4	Calcul de pose	81
4.1	État de l'art du calcul de pose en utilisant les moments	82
4.2	Pose par asservissement visuel virtuel	83
4.2.1	Asservissement visuel virtuel	83
4.2.2	Principe de la méthode proposée	84
4.2.3	Calcul de la rotation	85

4.2.4	Calcul de la translation	88
4.2.5	Résultats	88
4.3	Pose en utilisant une table de valeur	91
4.3.1	Principe de la méthode	91
4.3.2	Résultats	93
4.4	Application au cas discret	94
4.4.1	Pose par AVV	95
4.4.2	Pose en utilisant une table de valeur	98
4.5	Calcul partiel du déplacement 3D sans modèle de l'objet	100
4.5.1	État de l'art du calcul de pose sans modèle de l'objet	100
4.5.2	Méthode proposée	101
4.5.3	Résultats expérimentaux	103
4.5.3.1	Cas continu	103
4.5.3.2	Cas discret	104
4.6	Conclusion	106
5	Validation expérimentale en asservissement visuel	109
5.1	Choix des invariants r_i et r_j	110
5.2	Validation expérimentale: cas d'un objet plan et continu	124
5.2.1	Mouvement de translation pure	124
5.2.2	Mouvement complexe	130
5.2.3	Résultats avec une mauvaise calibration de la caméra et une occultation de l'objet	131
5.2.4	Cas d'un objet symétrique: un rectangle	139
5.2.5	Résultats expérimentaux avec les paramètres de la pose	142
5.3	Résultats expérimentaux: cas d'un objet discret	150
5.3.1	Mouvement de translation pure	150
5.3.2	Mouvement complexe	152
5.3.3	Résultats avec des erreurs de modélisation	154
5.3.4	Résultats en utilisant des images complexes	156
5.4	Conclusion	157
6	Conclusions et perspectives	159
6.1	Conclusion	159
6.2	Perspectives	162
A	Invariants aux mouvements de rotation	167
A.1	Moments 2D	170
A.2	Moments 3D	172
B	Minima, zone d'influence, ligne de partage des eaux	175
B.1	Définitions	175
B.1.1	Minima	175
B.1.2	Distance géodésique	175

B.1.3	Zone d'influence géodésique	175
B.1.4	Squelette par zone d'influence géodésique	176
Bibliographie		176

Chapitre 1

Introduction

1.1 Introduction générale

L'idée de créer des systèmes artificiels automatiques au service de l'homme est très ancienne. Cependant, ces créatures miraculeuses n'ont existé pendant plusieurs siècles que dans les légendes. En effet, les premières machines qui peuvent être considérées comme intelligentes sont apparues au début du siècle précédent, plus précisément dans le domaine agricole pour augmenter la production. Cette révolution a été suivie par une autre dans le domaine industriel, pour une bonne maîtrise de l'énergie et de la productivité. La fabrication des machines automatiques destinées à la production de masse a très vite été satisfaite par des manipulateurs dédiés à des tâches simples et bien précises. Pour se rendre polyvalent, les premiers robots sont apparus dans les années 60 aux USA. Toutefois, le rôle de ces robots était cantonné à des rôles d'automates mécaniques capables d'effectuer des trajectoires apprises à l'avance et répétées dans un environnement connu.

L'apparition des ordinateurs et l'évolution des capteurs extéroceptifs a poussé les chercheurs à tenter de rendre les robots plus autonomes dans des environnements susceptibles de varier. Divers capteurs peuvent être utilisés afin de permettre aux systèmes robotiques de pouvoir appréhender leur environnement. Les plus couramment utilisés sont les capteurs à ultrasons, les télémètres laser, les capteurs d'effort et les caméras. Le capteur à ultrasons a l'avantage de pouvoir être utilisé dans un environnement où la visibilité est faible mais il fournit une information très locale et de surcroît relativement imprécise. Il est souvent utilisé dans les applications de robotique mobile en association avec d'autres capteurs pour des tâches d'évitement d'obstacles. Le télémètre laser donne quant à lui une image très précise de l'environnement, mais il nécessite un balayage de l'espace, ce qui est très coûteux en temps. Il est souvent utilisé en extérieur dans des environnements totalement inconnus et difficilement modélisables où il permet de reconstruire la géométrie du terrain même si ce dernier est très peu marqué (c'est-à-dire des scènes peu riches en informations visuelles). Le capteur d'effort fournit également au robot un moyen d'appréhender un peu mieux son environnement. Il est souvent utilisé pour des applications de

montage, et plus particulièrement pour des tâches d'insertion ou de soudage. Il permet d'avoir une information sur les forces et les couples d'interaction entre le robot (ou une pièce qui est dans sa pince) et son environnement. Les capteurs visuels à leur tour fournissent une information riche qui permet aux robots de réaliser, de manière autonome, des tâches dans des environnements partiellement connus, ou totalement inconnus. L'utilisation d'informations visuelles dans les systèmes robotiques a souvent été fortement contrainte par les ressources de puissance de calcul. En effet, l'exploitation de la vision requiert une phase de traitement d'image qui est généralement coûteuse en temps de calcul. Ainsi, la période de mesure des premières applications de vision se comptait en dizaines de secondes (à la fin des années 1970). A cette époque, seule une exploitation en boucle ouverte de la mesure de vision était possible. La technique consistait à prendre une image de la scène, puis à la traiter, et enfin à déplacer le robot à l'endroit désiré en fonction de la mesure visuelle (cette approche est dénommée "static look and move" en anglais). Au cours de ces quinze dernières années, les progrès en matière de puissance de calcul des ordinateurs ont permis aux applications robotiques basées sur la vision de devenir de plus en plus "temps réel". Ainsi, il est désormais possible d'inclure la mesure extraite de l'image dans une boucle fermée, on parle alors d'asservissement visuel. Le schéma de contrôle consiste à annuler l'erreur entre la valeur des informations visuelles calculées pour l'état courant d'un système dynamique et pour son état désiré. Le système peut être réel (exemple d'une caméra montée sur l'organe terminal d'un robot) ou virtuel (cas de la réalité augmentée). Selon la nature des informations visuelles choisies, on peut distinguer plusieurs types d'asservissement visuel que nous présentons brièvement par la suite.

1.2 Asservissement visuel

Dans cette section, nous présentons successivement les différents principaux types d'asservissement visuel ainsi que leurs problèmes potentiels respectifs.

1.2.1 Asservissement visuel 2 D

Dans un asservissement visuel 2D, la référence est exprimée sous la forme d'informations visuelles exprimées directement dans l'image (voir figure 1.1). Le pionnier en matière d'asservissement 2D fut Weiss [Weiss 87]. Il étudia le lien entre le torseur cinématique de la caméra et les vitesses de déplacement des primitives dans l'image. Son travail fut développé entre autres par Feddema [Feddema 89], notamment du point de vue expérimental. Espiau et al. proposèrent un cadre plus formel dans [Chaumette 90, Espiau 92]. Un des principaux avantages de l'asservissement 2D est qu'il n'est pas nécessaire de connaître a priori un modèle de l'objet. Une approximation de la profondeur de l'objet par rapport à la caméra est généralement suffisante pour réaliser le schéma de commande. Ce schéma de commande est en

pratique robuste aux bruits de mesure et aux erreurs de calibration du système dans le cas où le déplacement à réaliser entre la position désirée du robot et sa position initiale est peu important. Cependant, en cas d'important déplacement à réaliser, ce type d'asservissement peut souffrir de plusieurs problèmes potentiels que nous décrirons de façon plus détaillée par la suite. En effet, si le comportement des primitives dans l'image est généralement satisfaisant étant donné que le contrôle est réalisé dans l'image, la trajectoire de la caméra peut être inadéquate dans le cas où de grands mouvements de rotation et de translation sont à réaliser. Le problème de convergence se pose également dans ces cas de figure.

Plusieurs types d'informations visuelles ont été proposés dans de précédents travaux. La plupart de ces travaux ont considéré des objets simples qui peuvent être exprimés par des primitives géométriques de type points, droites, ellipses, etc. Par conséquent, la famille des objets pouvant être traitée par ce type de méthodes était limitée. Par ailleurs, ce genre de méthodes nécessite la mise en correspondance entre les primitives dans un couple d'images (plus précisément entre l'image initiale et l'image désirée), ce qui n'est pas toujours facile à obtenir. D'autres méthodes ont été proposées pour surmonter les problèmes mentionnés auparavant en utilisant par exemple la projection dans l'espace propre [Deguchi 00] ou la signature du contour de l'objet [Collewet 00]. Le choix de primitives génériques pouvant décrire des objets de forme complexe reste tout de même un problème majeur, qui sera traité dans cette thèse.

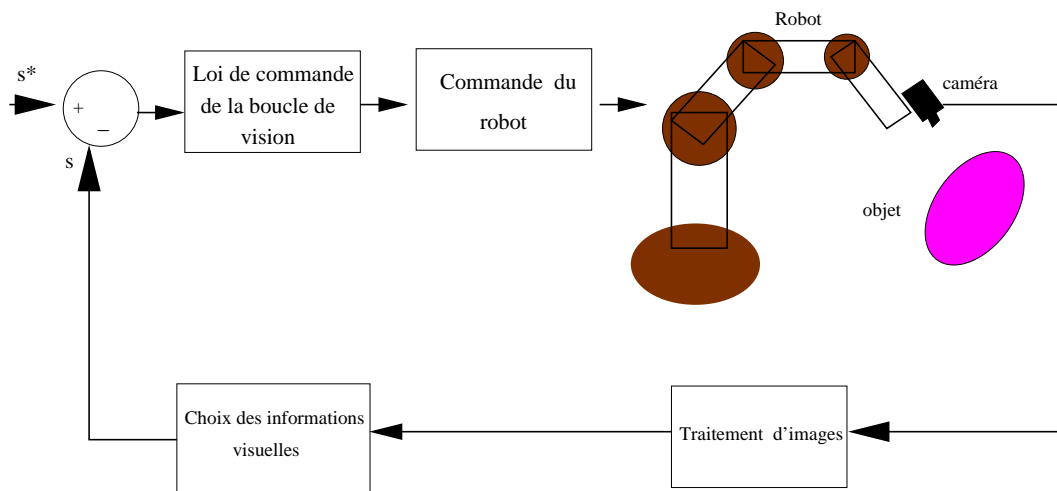


FIG. 1.1 – Principe d'un asservissement 2D

1.2.2 Asservissements visuels 3 D et 2 D 1/2

Dans un asservissement visuel 3D, la référence est exprimée sous la forme d'une pose à atteindre [Wilson 96] (voir figure 1.2). Cette pose définit la position et l'orientation d'un repère lié à l'objet par rapport à un repère lié à la caméra. La référence peut être également exprimée sous forme de primitives géométriques 3D [Martinet 96]. Le comportement dans le domaine 3D de ce type d'asservissement en cas de convergence du système est généralement satisfaisant du point de vue de la trajectoire de la caméra. Cependant, le calcul des informations 3D à utiliser dans la boucle de commande (pose ou primitives 3D) nécessite la connaissance du modèle 3D de l'objet. Ce type d'asservissement visuel présente également l'inconvénient d'être assez sensible aux erreurs de mesure. Le problème du maintien de l'objet dans le champ de vision de la caméra se pose aussi pour de grands déplacements étant donné que le contrôle se fait dans le domaine 3D sans prendre en compte ce qui se passe dans l'image. Un compromis des asservissements 3D et 2D a été réalisé en combinant ces deux types d'asservissement, par exemple via les schémas de commande 2D 1/2 proposés dans [Malis 99, Morel 00, Chaumette 00]. Ces schémas de commande utilisent une reconstruction partielle de la pose qui ne nécessite pas la connaissance du modèle 3D de l'objet. Ils permettent également d'obtenir un comportement intéressant dans le domaine 3D. Cependant, l'asservissement 2D 1/2 est moins robuste aux erreurs de mesure qu'un asservissement visuel 2D (surtout aux abords de la convergence où la reconstruction partielle est assez instable).

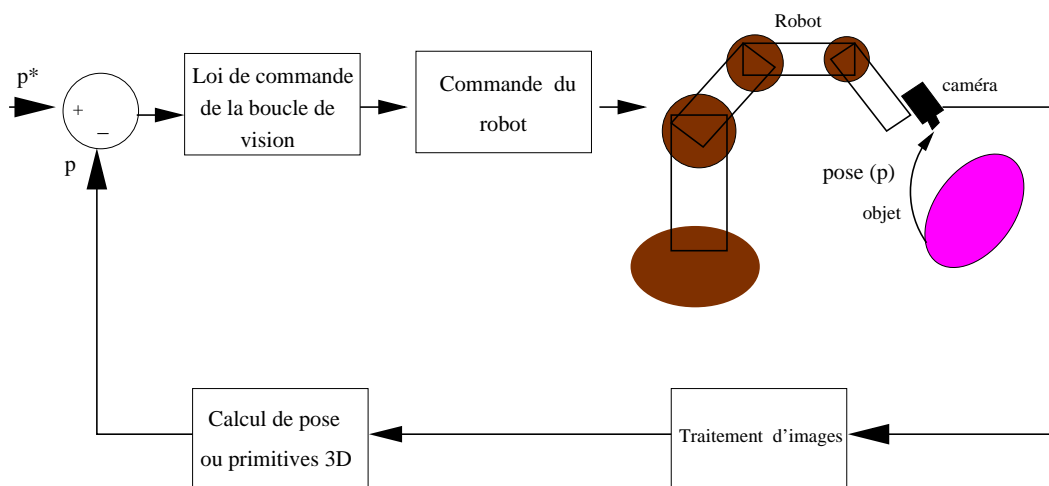


FIG. 1.2 – Principe d'un asservissement 3D.

1.2.3 Travaux récents en asservissement visuel

Récemment, plusieurs approches ont été proposées pour remédier aux problèmes posés des asservissements visuels classiques mentionnés précédemment. En effet, [Mezouar 03] propose une approche de génération de trajectoires dans l'image correspondant à des trajectoires optimales dans le domaine 3D. Les géodésiques et chemins les plus courts occupent logiquement une place privilégiée dans ce travail. Il est également possible de prendre en compte pendant la phase de génération de trajectoires dans l'image les contraintes de visibilité et d'occultation de l'objet [Mezouar 02]. Cependant, les trajectoires obtenus dans le domaine 3D ne sont plus forcément optimales. L'objectif général des ces approches était de palier le problème du comportement 3D d'un asservissement 2D tout en gardant les avantages de ce dernier, en particulier sa grande robustesse aux erreurs de modélisation par le suivi des trajectoires générées par asservissement visuel 2D. Cependant, seulement des primitives de type coordonnées de points ont été considérées dans ce travail. Les deux approches proposées étaient basées sur des fonctions de potentiel, qui peuvent théoriquement poser des problèmes de minima locaux. Une méthode similaire a été proposée dans [Cowan 02]. Cette méthode est basée sur l'utilisation de fonctions de navigation pour l'évitement d'obstacles et d'occultations de l'objet.

Récemment, une nouvelle approche basée sur le principe du "switching control" a été présentée dans [Chesi 02, Chesi 03] pour garder l'ensemble des points constituant l'objet dans le champ de vision de la caméra. Cette approche utilise un asservissement visuel 3D pour réaliser le déplacement de la caméra de sa position initiale à sa position désirée. Pour garder tous les points dans l'image, on commute sur d'autres lois de commande (typiquement un mouvement de recul) dans le cas où un point s'approche du bord ou risque de quitter l'image. Une autre approche basée sur le même principe du "switching control", mais cette fois en alternant asservissement 2D et asservissement 3D a été proposée dans [Gans 03].

Le choix d'informations visuelles pertinentes reste un des problèmes majeurs pour l'application des techniques d'asservissement visuel pour des objets complexes et dans le cas où de grands déplacements entre position initiale et position désirée sont à considérer. Le problème d'une représentation optimale et générique des informations visuelles se pose donc, aussi bien pour obtenir des propriétés satisfaisantes du point de vue de l'automatique (découplage, robustesse aux erreurs de mesure, absence de singularité, stabilité, comportement adéquat dans le domaine 3D, etc.), que pour pouvoir considérer avec la même représentation des objets de nature différente et de forme complexe.

1.3 Calcul de pose

Le calcul de pose consiste à déterminer la position et l'orientation d'un objet par rapport à la caméra en utilisant une image de cet objet et la connaissance de son modèle 3D. Comme nous l'avons mentionné précédemment, ce calcul sert de base à tout asservissement 3D. Par ailleurs, le calcul de pose a suscité beaucoup d'intérêts dans le domaine de la vision par ordinateur. En effet, parmi les applications qui font appel au calcul de pose, on trouve la réalité augmentée, la reconnaissance de formes, ou encore la mise en correspondance.

De nombreuses méthodes existent pour résoudre le problème du calcul de pose. Elles reposent sur la connaissance a priori d'un modèle 3D de l'objet et des paramètres de calibration de la caméra. Les primitives géométriques à la base du calcul de pose sont le plus souvent des points [Horaud 89, Haralick 89, Dementhon 95] ou des segments [Lowe 87, Dhome 89], voire des coniques [Safae-Rad 92, Ma 93], ou encore des objets cylindriques [Dhome 90]. Par contre, très peu de méthodes combinent des primitives de nature différente (voir cependant [Phong 95] pour l'utilisation conjointe de points et de droites). Les méthodes proposées dans la littérature pour le calcul de pose peuvent être également divisées en plusieurs catégories. Les méthodes sont soit purement géométriques [Horaud 89, Dhome 89], soit numériques et de type linéaire itératif [Dementhon 95] ou purement non linéaires [Lowe 87]. Sauf cas très particuliers [Horaud 89], on ne dispose pas d'une solution analytique à ce problème inverse. Les premières méthodes citées reposent sur la résolution de systèmes linéaires aux moindres carrés pour estimer la pose. Généralement, ces méthodes ne fournissent pas un résultat de très bonne qualité. Les techniques de minimisation non-linéaire consistent à minimiser directement l'erreur entre les observations dans l'image et la projection du modèle de l'objet pour une pose donnée. La minimisation est généralement réalisée en utilisant des algorithmes numériques itératifs de type Newton-Raphson ou Levenberg-Marquardt. Le principal avantage de ces approches est la précision du résultat obtenu. En contrepartie, l'algorithme de minimisation est sensible aux minima locaux, et peut, dans certains cas critiques, diverger. C'est pourquoi une bonne initialisation du vecteur de paramètres à estimer est souvent nécessaire.

Le problème du calcul de pose peut être vu également comme le problème dual de l'asservissement visuel [Sundareswaran 98, Marchand 02]. Ces méthodes considèrent le problème du calcul de pose comme le problème du positionnement d'une caméra virtuelle en utilisant un asservissement 2D. Le principe consiste à déplacer cette caméra virtuelle d'une position initiale connue vers la position de la caméra référencée par la pose qu'on désire calculer en minimisant l'erreur entre les informations extraites de l'image acquise dans la position inconnue de la caméra et celles calculées pour la position de la caméra virtuelle. Le déplacement de la caméra virtuelle est alors calculé en utilisant les mêmes lois de commande que dans le cas d'un asservissement visuel. Nous reviendrons sur le principe de cette méthode avec plus de détails dans le chapitre 4 de ce manuscrit.

On peut déjà signaler que, comme pour l'asservissement visuel et quelle que soit la méthode utilisée pour le calcul de pose, le choix d'informations optimales reste un problème majeur. Dans cette thèse, nous proposons l'utilisation des moments comme informations visuelles pour résoudre les problèmes posés en asservissement visuel et en calcul de pose.

1.4 Les moments comme informations visuelles

Les moments sont des informations génériques pouvant décrire des objets de forme complexe et qui sont définis par une surface continue, un volume, un ensemble de points ou encore une courbe fermée. Par ailleurs, il est possible à partir des moments de dériver des fonctions invariantes à certaines transformations que peut subir l'objet considéré. Ces fonctions sont appelées moments invariants. Cette propriété intéressante, qui a donné naissance à la théorie des moments invariants, occupe une place importante dans le domaine de la vision par ordinateur. Elle a été introduite pour la première fois en 1962 par Hu [Hu 62]. Ce dernier a établi le théorème fondamental des moments invariants (FTMI) en se basant sur la théorie des invariants algébriques. Ce théorème a été utilisé pour dériver les sept fameux invariants de Hu. Depuis 1962, les moments invariants ont été utilisés dans plusieurs domaines : calcul de pose [Cyganski 85] [Mukundan 96], reconnaissance de caractères [Wong 95], contrôle qualité dans le domaine industriel [Sluzek 95], mise en correspondance [Wong 78], fusion de données multi-capteurs [Markandey 92], etc.

Dans ce travail, nous proposons d'utiliser les moments en asservissement visuel et en calcul de pose. Ce choix permettra de traiter des objets de forme complexe sans nécessiter de phase d'appariement si un seul objet est visible. De plus, la propriété d'invariance de certaines fonctions de moments vis-à-vis de certains mouvements et transformations sera exploitée pour réaliser un schéma de commande découplé en asservissement visuel 2D. Ce découplage permettra alors d'éviter toute singularité à la tâche. Nous utiliserons également les moments pour minimiser les non-linéarités de la matrice d'interaction reliant la vitesse des primitives et les vitesses appliquées à la caméra. Cela permettra, comme nous l'expliquerons dans la suite de ce manuscrit, d'obtenir des comportements adéquats dans le domaine 3D.

1.5 Organisation du manuscrit

La suite de ce manuscrit est organisée comme suit :

- Dans le chapitre 2, nous rappelons les définitions classiques des moments et les notions de base de la théorie des invariants algébriques. Nous proposons également une nouvelle méthode générale et systématique du calcul des invariants aux mouvements de rotations et aux changements d'échelle. La méthode que nous proposons est valable pour un espace de dimension quelconque. Nous proposons également une méthode pour éliminer facilement la redondance entre les invariants obtenus.
- Au début du chapitre 3, nous donnons un état de l'art de l'asservissement visuel et nous décrivons ses problèmes potentiels. Nous déterminons également la forme analytique de la matrice d'interaction associée aux moments calculés à partir d'une image d'un objet de forme plane. Cette matrice joue un rôle essentiel dans l'élaboration de la loi de commande en asservissement visuel. Les cas de la projection perspective et de la projection sphériques sont considérés. Finalement, nous utilisons les résultats obtenus dans le chapitre 2 pour construire six combinaisons de moments permettant de réaliser un schéma de commande découplé et minimisant les non-linéarités de la matrice d'interaction. Le cas où la position désirée de l'objet est parallèle au plan image de la caméra est tout d'abord étudié. Nous généralisons par la suite les résultats obtenus au cas où la position désirée de l'objet n'est pas parallèle au plan image.
- Dans le chapitre 4, nous proposons deux méthodes basées sur les moments invariants pour résoudre le problème du calcul de pose. La première méthode utilise un asservissement visuel virtuel. La seconde méthode est basée sur le calcul d'une table de valeur pour la pose.
- Dans le chapitre 5, nous validons les résultats théoriques obtenus dans le chapitre 3 par des résultats expérimentaux obtenus avec un système robotique à six degrés de liberté et pour des objets plans de forme complexe.

Une conclusion et deux annexes terminent le document. Nous résumons dans cette conclusion les principaux résultats obtenus et évoquons quelques perspectives de recherche.

Chapitre 2

Les moments invariants

Dans ce chapitre, nous proposons une nouvelle méthode de calcul des moments invariants dans un espace de dimension quelconque (invariants aux mouvements de translations, aux rotations et aux changements d'échelle). Nous allons traiter dans un premier temps le cas des moments d'une distribution continue (que nous appellerons par la suite **Cas continu**). Ensuite, nous généraliserons les résultats obtenus au cas des moments d'une distribution discrète, c'est-à-dire construits à partir d'un ensemble de points (ce cas sera appelé **Cas discret**).

2.1 Définitions

Dans cette section, nous rappelons quelques définitions sur les moments. Nous nous intéressons, ici, seulement au moments cartésiens. D'autres formes de moments peuvent être définies (moments de Zernike, moments complexes,...), mais ils ne présentent pas un intérêt particulier pour les applications traitées dans ce travail. Plus de détails sur ces moments peuvent être trouvés dans [Mukundan 98, Prokop 92].

2.1.1 Moments 2D : définitions et propriétés

Les moments 2D m_{pq} d'ordre $p + q$ d'une fonction de densité $f(x, y)$ sont définis par :

$$m_{pq} \equiv \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy \quad (2.1)$$

où $f(x, y)$ est le niveau de gris d'un point de coordonnées (x, y) dans le cas d'une image.

• **Moment d'ordre zéro : surface.** Le moment d'ordre zéro m_{00} d'une distribution $f(x, y)$ est donné par :

$$m_{00} \equiv \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) dx dy \quad (2.2)$$

Il représente la masse totale de la distribution $f(x, y)$. Dans le cas d'un objet binaire, m_{00} n'est autre que la surface de l'objet.

• **Moments d'ordre 1 : centre de gravité.** Les moments d'ordre 1, c'est-à-dire m_{10} et m_{01} , permettent de déterminer la position du centre de gravité de l'objet. Plus précisément, si on note x_g et y_g les coordonnées du centre de gravité de l'objet, on a :

$$x_g = \frac{m_{10}}{m_{00}} \text{ et } y_g = \frac{m_{01}}{m_{00}}. \quad (2.3)$$

• **Moments centrés :** les moments centrés μ_{pq} sont calculés par rapport au centre de gravité de l'objet (x_g, y_g) . Ils sont définis par :

$$\mu_{pq} \equiv \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - x_g)^p (y - y_g)^q f(x, y) dx dy \quad (2.4)$$

Les moments centrés peuvent être calculés directement à partir des moments m_{pq} d'ordres inférieurs à $p + q + 1$ par :

$$\mu_{pq} = \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} (-x_g)^{p-k} (-y_g)^{q-l} m_{kl} \quad (2.5)$$

Réciproquement, il est possible de déterminer les moments à partir des moments centrés. On a en effet :

$$m_{pq} = \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} (x_g)^{p-k} (y_g)^{q-l} \mu_{kl} \quad (2.6)$$

On peut dès à présent signaler que les moments 2D centrés sont invariants aux translations 2D.

• **Moments centrés d'ordre 2 : axes principaux.** Les moments centrés d'ordre 2 μ_{20} , μ_{11} , et μ_{02} sont appelés les moments d'inertie. Avec les coordonnées du centre de gravité, ils permettent de définir une ellipse dont les moments d'ordre inférieur ou égal à 2 ont la même valeur que ceux de l'objet (voir Figure 2.1). L'équation de cette ellipse est donnée par :

$$\frac{(x - x_g + t(y - y_g))^2}{a_1^2(1 + t^2)} + \frac{(y - y_g - t(x - x_g))^2}{a_2^2(1 + t^2)} - 1 = 0 \quad (2.7)$$

où a_1 et a_2 représentent respectivement les deux axes principaux majeur et mineur et où t est liée à l'orientation de l'objet. Plus précisément, on a $t = \tan \alpha$ où :

$$\alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \quad (2.8)$$

À partir des paramètres de l'ellipse, on peut facilement exprimer les moments centrés d'ordre 2. On a en effet :

$$\begin{cases} \mu_{20} = m_{00}((a_1^2 + a_2^2 t^2)/4(1 + t^2)) \\ \mu_{02} = m_{00}((a_1^2 t^2 + a_2^2)/4(1 + t^2)) \\ \mu_{11} = m_{00}(t(a_1^2 - a_2^2)/4(1 + t^2)) \end{cases} \quad (2.9)$$

où $m_{00} = \pi a_1 a_2$.

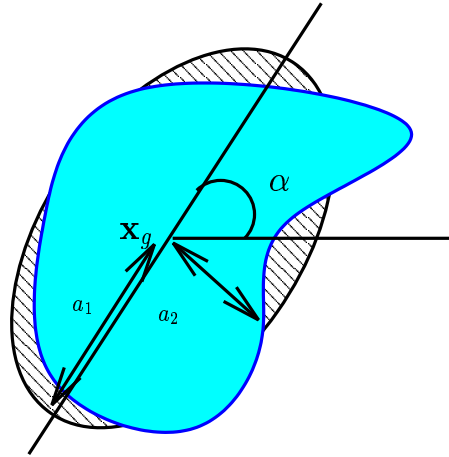


FIG. 2.1 – Orientation d'un objet et définition des paramètres usuels d'une ellipse

Réciproquement, les paramètres de l'ellipse peuvent être déterminés à partir des équations suivantes :

$$\begin{cases} a_1^2 = \frac{2(\mu_{02} + \mu_{20} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})}{m_{00}} \\ a_2^2 = \frac{2(\mu_{02} + \mu_{20} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})}{m_{00}} \\ t = (\mu_{02} - \mu_{20} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})/2\mu_{11} \end{cases}$$

• **Moments centrés d'ordre 3 : asymétrie.** Les moments centrés d'ordre 3 évaluent le degré d'asymétrie d'un objet par rapport à ses axes principaux. En

reconnaissance des formes, on utilise souvent les coefficients d'asymétrie définis de la manière suivante :

$$S_x = \frac{\mu_{30}}{\mu_{20}^{3/2}} \text{ et } S_y = \frac{\mu_{03}}{\mu_{02}^{3/2}} \quad (2.10)$$

Les moments d'ordre 3 permettent également de résoudre le problème du choix de la valeur de α . En effet, on peut noter à partir de la figure 2.1 qu'il y a deux solutions pour l'orientation de l'objet : α et $\alpha + \pi$. Les moments d'ordre 3, lorsqu'ils ne sont pas tous nuls, peuvent être employés pour résoudre cette ambiguïté. En effet, une rotation d'un objet de 180 degrés change le signe des moments d'ordre 3. Pour distinguer les deux solutions de α , on peut fixer le sens des axes principaux dans une image de référence. Supposons alors que $\mu_{r_{ij}}$ soient les moments d'ordre 3 calculés pour l'image de référence dans un repère dont le centre coïncide avec le centre de l'objet et α_r la valeur de l'orientation de l'objet fixée pour cette image. Les moments après un mouvement de rotation peuvent être calculés par la formule suivante :

$$\mu'_{ij} = \sum_{l_1=0}^i \sum_{l_2=0}^j \binom{i}{l_1} \binom{j}{l_2} r_{11}^{l_1} r_{21}^{l_2} r_{12}^{i-l_1} r_{22}^{j-l_2} \mu_{l_1+l_2, i+j-(l_1+l_2)} \quad (2.11)$$

ou r_{mn} sont les éléments de la matrice de rotation dans le plan d'angle $(\alpha - \alpha_r)$:

$$\begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} \cos(\alpha - \alpha_r) & \sin(\alpha - \alpha_r) \\ -\sin(\alpha - \alpha_r) & \cos(\alpha - \alpha_r) \end{bmatrix} \quad (2.12)$$

En effet, on a :

$$\mu'_{ij} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (r_{11}x + r_{12}y)^i (r_{21}x + r_{22}y)^j f(r_{11}x + r_{12}y, r_{21}x + r_{22}y) dx dy \quad (2.13)$$

Si on suppose que $f(r_{11}x + r_{12}y, r_{21}x + r_{22}y) = f(x, y)$ (c'est-à-dire que la fonction de densité en un point de l'image ne change pas après rotation), on obtient (2.11) après développement de $(r_{11}x + r_{12}y)^i (r_{21}x + r_{22}y)^j$.

L'orientation de l'objet dans une image donnée est finalement choisie telle que les moments d'ordre 3 calculés après la rotation de $\alpha - \alpha_r$ autour du centre de l'objet en utilisant (2.11) aient les mêmes signes que leurs correspondants $\mu_{r_{ij}}$ calculés pour l'image de référence.

• **Moments centrés d'ordre 4: Kurtosis de l'objet.** Les moments centrés d'ordre 4 quant-à eux expriment le Kurtosis de l'objet, c'est-à-dire le degré d'aplatissement de celui-ci par rapport à une distribution normale selon une direction donnée (l'axe x ou l'axe y). Ces deux entités sont définies par :

$$K_x = \frac{\mu_{40}}{\mu_{20}^2} - 3 \text{ et } K_y = \frac{\mu_{04}}{\mu_{02}^2} - 3 \quad (2.14)$$

Pour une distribution normale, le Kurtosis est égal à 0. Un Kurtosis inférieur à zéro indique que la distribution de l'objet selon la direction considérée est plus plate qu'une distribution normale. Nous rappelons que l'équation de la courbe normale est donnée par (voir figure 2.2) :

$$p(x) = \frac{1}{\pi\sigma} \exp \left[-\frac{1}{2} \left(\frac{x - x_g}{\sigma} \right)^2 \right] \quad (2.15)$$

où x_g est la moyenne de la distribution normale et σ son écart type. Lorsque le Kur-

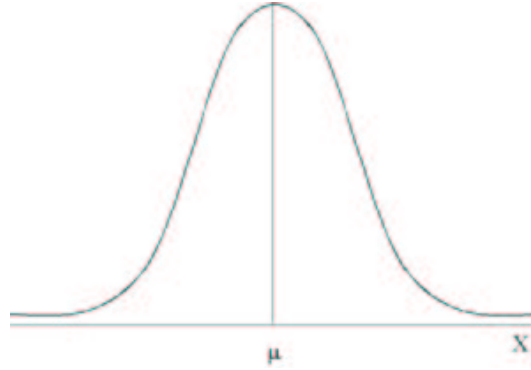


FIG. 2.2 – Courbe d'une distribution normale

tosis de l'objet selon une direction choisie est supérieur à 0, l'objet est plus incurvé qu'une distribution normale selon la même direction. Par exemple, le Kurtosis d'une distribution exponentielle est égal à 6.

• **Moments d'ordre supérieur.** Malheureusement, les moments d'ordres supérieurs sont encore moins intuitifs, et n'ont pas d'interprétation géométrique aisée.

2.1.2 Moments 3D et moments k-dimensionnels

- Les moments m_{pqr} d'ordre $p + q + r$ de la fonction de densité $f(x, y, z)$ sont définis de la même manière que les moments 2D :

$$m_{pqr} \equiv \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q z^r f(x, y, z) dx dy dz \quad (2.16)$$

- Les moments 3D centrés μ_{pqr} sont définis par :

$$\mu_{pqr} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - x_g)^p (y - y_g)^q (z - z_g)^r f(x, y, z) dx dy dz \quad (2.17)$$

où $x_g = \frac{m_{100}}{m_{000}}$, $y_g = \frac{m_{010}}{m_{000}}$, et $z_g = \frac{m_{001}}{m_{000}}$ sont les coordonnées du centre de gravité de l'objet.

– Par analogie, les moments d'une fonction de densité $f(x_1, \dots, x_k)$ sont définis par :

$$m_{p_1 \dots p_k} \equiv \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} x_1^{p_1} \dots x_k^{p_k} f(x_1, \dots, x_k) dx_1 \dots dx_k \quad (2.18)$$

où $p = p_1 + \dots + p_k$ est l'ordre du moment $m_{p_1 \dots p_k}$. Finalement, les moments centrés k -dimensionnels sont définis par :

$$\mu_{p_1 \dots p_k} \equiv \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} (x_1 - x_{1g})^{p_1} \dots (x_k - x_{kg})^{p_k} f(x_1, \dots, x_k) dx_1 \dots dx_k \quad (2.19)$$

avec $x_{1g} = \frac{m_{10 \dots 0}}{m_{0 \dots 0}}$, \dots , $x_{kg} = \frac{m_{0 \dots 0k}}{m_{0 \dots 0}}$. Ils peuvent être calculés à partir des moments donnés par l'équation (2.18) par :

$$\mu_{p_1 \dots p_k} = \sum_{i_1=0}^{p_1} \dots \sum_{i_k=0}^{p_k} (-1)^{i_1 + \dots + i_k} \binom{p_1}{i_1} \dots \binom{p_k}{i_k} x_{1g}^{i_1} \dots x_{kg}^{i_k} m_{p_1 - i_1 \dots p_k - i_k} \quad (2.20)$$

Réciproquement, les moments peuvent être calculés à partir des moments centrés par :

$$m_{p_1 \dots p_k} = \sum_{i_1=0}^{p_1} \dots \sum_{i_k=0}^{p_k} \binom{p_1}{i_1} \dots \binom{p_k}{i_k} x_{1g}^{i_1} \dots x_{kg}^{i_k} \mu_{p_1 - i_1 \dots p_k - i_k} \quad (2.21)$$

Les moments centrés sont invariants aux mouvements de translations dans leur espace de dimension k , c'est-à-dire qu'ils sont invariants à la transformation suivante :

$$\begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} x'_1 \\ \vdots \\ x'_k \end{bmatrix} + \begin{bmatrix} T_1 \\ \vdots \\ T_k \end{bmatrix}, \forall T_1, \dots, T_k \quad (2.22)$$

Finalement, on peut rappeler que les moments d'une fonction de densité f existent si f est continue par morceaux et non nulle uniquement sur des régions finies de l'espace.

2.2 Calcul des moments invariants

2.2.1 Introduction

La théorie des moments invariants a été introduite pour la première fois en 1962 par Hu [Hu 62]. Celui-ci a établi le théorème fondamental des moments 2D invariants (FTMI) en se basant sur la théorie des invariants algébriques. Ce théorème a été utilisé pour dériver sept fameux invariants 2D aux mouvements de rotation :

$$\begin{aligned}
I_1 &= \mu_{20} + \mu_{02} \\
I_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\
I_3 &= (\mu_{30} - 3\mu_{12})^2 + (\mu_{03} - 3\mu_{21})^2 \\
I_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{03} + \mu_{21})^2 \\
I_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{03} + \mu_{21})^2) \\
&\quad + (3\mu_{21} - \mu_{03})(\mu_{03} + \mu_{21})(3(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2) \\
I_6 &= (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} - \mu_{03})^2) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \\
I_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{03} + \mu_{21})^2) \\
&\quad - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})((3\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2)
\end{aligned} \tag{2.23}$$

En fait, le FTMI contient quelques erreurs, comme cela a été démontré par Mamistvalov en 1970 ([Mamistvalov 70] publié en russe). Reiss a ensuite établi une version corrigée du FTMI (RFTMI) dans le cas 2D [Reiss 91]. Le FTMI a été généralisé au cas des moments définis dans un espace de dimension k et appliqué dans le domaine de la fusion de données multi-capteurs dans [Markandey 92]. Malheureusement, ce théorème contient la même erreur que le FTMI proposé dans [Hu 62]. Finalement, Mamistvalov a donné la bonne généralisation du RFTMI dans un espace de dimension k [Mamistvalov 98]. Bien que le RFTMI soit valable dans un espace de dimension k , il ne permet pas, comme on pourra le constater à partir de son énoncé (voir paragraphe 2.2.2.2), une détermination systématique des moments invariants.

Dans la littérature, plusieurs techniques non basées sur le RFTMI traitent du problème de la détermination des moments invariants. Dans le cas d'un espace de dimension 2 : Reddi [Reddi 81] a proposé une méthode de calcul des moments invariants à partir des moments radiaux et angulaires, Teague [Teague 80], Belkassim [Belkassim 90], et Walin [Walin 95] ont obtenu des moments invariants à partir des moments de Zernike, Abu Mostapha [Abu-Mustapha 85] et Flusser [Flusser 00] ont obtenu des moments invariants aux mouvements de rotation à partir des moments complexes. Dans le cas des moments définis dans un espace de dimension 3, Chong [Lo 89] a déterminé des moments invariants aux rotations en utilisant la représentation en harmoniques sphériques. Cependant, il n'existe pas une méthode systématique valable dans un espace de dimension k .

Quelle que soit la méthode utilisée pour obtenir des moments invariants, ces derniers peuvent être dépendants les uns des autres. En effet, la somme, le produit et le rapport de deux invariants à une transformation donnée sont aussi des

invariants à la même transformation. Pour éliminer toute redondance inutile, il est nécessaire d'établir une règle pour construire une base d'invariants indépendants. Dans la littérature, un seul travail [Flusser 00] a traité ce problème de la redondance des invariants en utilisant les moments complexes.

Dans la suite de ce chapitre, nous proposons tout d'abord une nouvelle méthode pour calculer des moments invariants (invariants aux rotations, aux translations, et aux changements d'échelle). À partir de cette méthode, nous établirons ensuite une règle simple pour construire une base de moments invariants et éliminer ainsi la redondance entre invariants.

2.2.2 Théorie des invariants algébriques

2.2.2.1 Concepts

Dans ce paragraphe, nous rappelons quelques concepts sur la théorie des invariants algébriques nécessaires pour la compréhension du RTFMI. Plus de détails sur ces concepts sont donnés dans [Guerivich 64, Hilbert 93].

• **Polynôme homogène.** Le polynôme d'ordre p de k variables $x_1 \dots x_k$:

$$H(x_1, \dots, x_k) = \sum_{p_1 + \dots + p_k = p} \frac{p!}{p_1! \dots p_k!} a_{p_1 \dots p_k} x_1^{p_1} \dots x_k^{p_k} \quad (2.24)$$

est appelé forme algébrique k -naire d'ordre p . Pour $k = 2$, H est appelé forme algébrique binaire d'ordre p . Par exemple, $H(x_1, x_2) = a_{20}x_1^2 + 2a_{11}x_1x_2 + a_{02}x_2^2$ est la forme algébrique binaire d'ordre 2.

• **Invariant algébrique relatif.** Soit $a = (a_{p \dots 0} \dots a_{0 \dots p})$ le vecteur de paramètres qui intervient dans la forme polynomiale (2.24). Un polynôme homogène $I(a)$ d'ordre n est un invariant algébrique relatif de poids w si et seulement si :

$$I(a') = \Delta^w I(a) \quad (2.25)$$

où a' est le vecteur de paramètres qui intervient dans la forme polynomiale après la transformation suivante :

$$\begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} = \mathbf{R} \begin{bmatrix} x'_1 \\ \vdots \\ x'_k \end{bmatrix} \quad (2.26)$$

et où Δ est égal au déterminant de la matrice \mathbf{R} . Si $w = 0$, le polynôme $I(a)$ est appelé invariant absolu.

Par exemple, si on considère la forme polynomiale binaire suivante :

$$f(x_1, x_2) = a_{20}x_1^2 + a_{11}x_1x_2 + a_{02}x_2^2 \quad (2.27)$$

le polynôme :

$$I(a) = a_{20}a_{02} - a_{11}^2 \quad (2.28)$$

est un invariant relatif d'ordre $n = 2$ et de poids $w = 2$.

Preuve :

La forme algébrique donnée par (2.27) peut être écrite sous la forme :

$$H(\mathbf{X}) = \mathbf{X}^T \mathbf{A} \mathbf{X}$$

où $\mathbf{X} = (x_1, x_2)$ et :

$$\mathbf{A} = \begin{bmatrix} a_{20} & a_{11} \\ a_{11} & a_{02} \end{bmatrix}$$

Après la transformation (2.26), on obtient :

$$H(\mathbf{X}') = \mathbf{X}'^T \mathbf{A}' \mathbf{X}'$$

où :

$$\mathbf{A}' = \mathbf{R}^T \mathbf{A} \mathbf{R}$$

On a $\det(\mathbf{R}) = \det(\mathbf{R}^T) = \Delta$ et donc $I(a') = \det(\mathbf{A}') = \Delta^2 \det(\mathbf{A}) = \Delta^2 I(a)$.

2.2.2.2 Le théorème fondamental des moments invariants révisé

Théorème 1 Soit $H(x_1, \dots, x_k)$ une forme algébrique k - naire d'ordre p .

Si $H(x_1, \dots, x_k)$ a un invariant algébrique d'ordre n et de poids w :

$$I(a') = \Delta^w I(a) \quad (2.29)$$

alors les moments d'ordre p ont le même invariant, mais de poids $w + n$:

$$I(m'_{p\dots 0}, \dots, m'_{0\dots p}) = \Delta^{w+n} I(m_{p\dots 0}, \dots, m_{0\dots p}) \quad (2.30)$$

La démonstration détaillée de ce théorème est donnée dans [Reiss 91] pour le cas 2D et dans [Mamistvalov 98] dans le cas général d'un espace de dimension k . Comme exemple d'application de ce théorème, considérons la forme algébrique donnée par (2.27). Le polynôme donné par (2.28) est un invariant relatif d'ordre 2 et de poids $w = 2$. On en déduit :

$$I(m'_{20}, m'_{11}, m'_{02}) = \Delta^4 I(m_{20}, m_{11}, m_{02}) \quad (2.31)$$

soit encore :

$$m'_{20}m'_{02} - m'_{11}{}^2 = \Delta^4(m_{20}m_{02} - m_{11}^2) \quad (2.32)$$

La combinaison de moments $m_{02}m_{02} - m_{11}^2$ est donc un invariant d'ordre 2 et de poids 4 à la transformation \mathbf{R} . On peut ainsi noter que les moments invariants peuvent être obtenus à partir des invariants algébriques. Cependant, il n'existe pas de règle générale pour obtenir ces derniers. Dans ce qui suit, nous allons proposer une méthode générale pour la détermination de ces moments invariants.

2.2.3 Méthode générale de calcul des moments invariants

Dans la suite de ce paragraphe, nous nous intéressons aux moments invariants aux mouvements de rotation et aux changements d'échelle. Pour avoir l'invariance aux mouvements de translation, il suffit de considérer les moments centrés comme nous l'avons déjà vu. La méthode que nous proposons est basée sur une approche variationnelle similaire à celle proposée par Van Gool en utilisant l'algèbre de Lie [VanGool 95].

2.2.3.1 Invariants aux transformations orthogonales

Une transformation orthogonale, typiquement une rotation, peut s'écrire sous la forme :

$$\mathbf{X}(t) = \mathbf{R}\mathbf{X}(t_1) \quad (2.33)$$

où $\mathbf{R}\mathbf{R}^T = \mathbf{I}_k$. Après dérivation de (2.33) par rapport au temps, on obtient :

$$\dot{\mathbf{X}}(t_1) = \mathbf{A}\mathbf{X}(t_1) \quad (2.34)$$

où $\mathbf{A} = \mathbf{R}^T \dot{\mathbf{R}}$ est une matrice antisymétrique. Dans une première étape, on cherchera des polynômes invariants d'ordre 1, soit tels que :

$$q(\mathbf{m}_p^1) = \boldsymbol{\alpha}^T \mathbf{m}_p^1 \quad (2.35)$$

où $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_l)$ est un vecteur de paramètres à calculer, et $l = \dim(\mathbf{m}_p^1)$. Le polynôme $q(\mathbf{m}_p^1)$ est invariant à (2.33), si et seulement si :

$$\dot{q}(\mathbf{m}_p^1) = \boldsymbol{\alpha}^T \dot{\mathbf{m}}_p^1 = 0, \quad \forall \mathbf{m}_p^1 \quad (2.36)$$

Nous allons donc déterminer la forme de $\dot{m}_{p_1 \dots p_k}$ pour exhiber une condition sur $\boldsymbol{\alpha}$. Pour cela, nous considérons tout d'abord le cas général d'une transformation quelconque, puis nous appliquons le résultat trouvé à une rotation. On a par définition :

$$\dot{m}_{p_1 \dots p_k}(t_1) = \lim_{t \rightarrow t_1} \frac{m_{p_1 \dots p_k}(t) - m_{p_1 \dots p_k}(t_1)}{t - t_1} \quad (2.37)$$

où :

$$m_{p_1 \dots p_k}(t) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} x_1^{p_1}(t) \dots x_k^{p_k}(t) f(\mathbf{X}(t)) d\mathbf{X}(t) \quad (2.38)$$

En appliquant un changement de variable, l'équation (2.38) peut se réécrire sous la forme :

$$m_{p_1 \dots p_k}(t) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} x_1^{p_1}(t) \dots x_k^{p_k}(t) f(\mathbf{X}(t)) \Delta_k(t) d\mathbf{X}(t_1) \quad (2.39)$$

avec :

$$\Delta_k(t) = \begin{vmatrix} \frac{\partial x_1(t)}{\partial x_1(t_1)} & \cdots & \frac{\partial x_1(t)}{\partial x_k(t_1)} \\ \vdots & & \vdots \\ \frac{\partial x_k(t)}{\partial x_1(t_1)} & \cdots & \frac{\partial x_k(t)}{\partial x_k(t_1)} \end{vmatrix} \quad (2.40)$$

On peut remarquer qu'on a évidemment $\Delta_k(t_1) = |\mathbf{I}_k| = 1$. A présent, soit $h(t)$ la fonction définie par :

$$h(t) = x_1^{p_1}(t) \dots x_k^{p_k}(t) f(\mathbf{X}(t)) \Delta_k(t) \quad (2.41)$$

En combinant (2.41) et (2.37), on obtient :

$$\begin{aligned} \dot{m}_{p_1 \dots p_k}(t_1) &= \lim_{t \rightarrow t_1} \frac{1}{t - t_1} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} [h(t) - h(t_1)] d\mathbf{X}(t_1) \\ &= \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \dot{h}(t_1) d\mathbf{X}(t_1) \end{aligned} \quad (2.42)$$

En dérivant l'expression de h , on obtient :

$$\begin{aligned} \dot{h}(t) &= \frac{d[x_1^{p_1}(t) \dots x_k^{p_k}(t)]}{dt} f(\mathbf{X}(t)) \Delta_k(t) + x_1^{p_1}(t) \dots x_k^{p_k}(t) \frac{d[f(\mathbf{X}(t))]}{dt} \Delta_k(t) \\ &\quad + x_1^{p_1}(t) \dots x_k^{p_k}(t) f(\mathbf{X}(t)) \frac{d[\Delta_k(t)]}{dt} \end{aligned} \quad (2.43)$$

En combinant (2.43) et (2.42), on obtient :

$$\begin{aligned} \dot{m}_{p_1 \dots p_k}(t_1) &= \sum_{i=1}^k \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} p_i x_i^{p_i-1}(t_1) \dot{x}_i(t_1) \prod_{j=1, j \neq i}^k x_j^{p_j}(t_1) f(\mathbf{X}(t_1)) d\mathbf{X}(t_1) \\ &\quad + \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} x_1^{p_1} \dots x_k^{p_k} \dot{f}(\mathbf{X}(t_1)) d\mathbf{X}(t_1) \\ &\quad + \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} x_1^{p_1}(t_1) \dots x_k^{p_k}(t_1) f(\mathbf{X}(t_1)) \left(\frac{d[\Delta_k(t)]}{dt} (t_1) \right) d\mathbf{X}(t_1) \end{aligned} \quad (2.44)$$

Considérons tout d'abord le cas 2D ($k = 2$), l'équation (2.40) peut s'écrire :

$$\Delta_2(t) = \begin{vmatrix} \frac{\partial x_1(t)}{\partial x_1(t_1)} & \frac{\partial x_1(t)}{\partial x_2(t_1)} \\ \frac{\partial x_2(t)}{\partial x_1(t_1)} & \frac{\partial x_2(t)}{\partial x_2(t_1)} \end{vmatrix} = \frac{\partial x_1(t)}{\partial x_1(t_1)} \frac{\partial x_2(t)}{\partial x_2(t_1)} - \frac{\partial x_2(t)}{\partial x_1(t_1)} \frac{\partial x_1(t)}{\partial x_2(t_1)} \quad (2.45)$$

Après dérivation de (2.45), on obtient :

$$\dot{\Delta}_2(t) = \frac{\partial \dot{x}_1(t)}{\partial x_1(t_1)} \frac{\partial x_2(t)}{\partial x_2(t_1)} + \frac{\partial x_1(t)}{\partial x_1(t_1)} \frac{\partial \dot{x}_2(t)}{\partial x_2(t_1)} - \left(\frac{\partial \dot{x}_2(t)}{\partial x_1(t_1)} \frac{\partial x_1(t)}{\partial x_2(t_1)} + \frac{\partial x_2(t)}{\partial x_1(t_1)} \frac{\partial \dot{x}_1(t)}{\partial x_2(t_1)} \right) \quad (2.46)$$

Puisque $\Delta_2(t_1) = |\mathbf{I}_2|$, on a $\frac{\partial x_1(t_1)}{\partial x_1(t_1)} = \frac{\partial x_2(t_1)}{\partial x_2(t_1)} = 1$ et $\frac{\partial x_2(t_1)}{\partial x_1(t_1)} = \frac{\partial x_1(t_1)}{\partial x_2(t_1)} = 0$, d'où on déduit :

$$\dot{\Delta}_2(t_1) = \frac{\partial \dot{x}_1}{\partial x_1}(t_1) + \frac{\partial \dot{x}_2}{\partial x_2}(t_1) \quad (2.47)$$

Pour $k > 2$, de la même façon que précédemment, on peut obtenir :

$$\dot{\Delta}_k(t_1) = \frac{\partial \dot{x}_1}{\partial x_1}(t_1) + \frac{\partial \dot{x}_2}{\partial x_2}(t_1) + \dots + \frac{\partial \dot{x}_k}{\partial x_k}(t_1) \quad (2.48)$$

Dans le cas d'une transformation orthogonale, la matrice \mathbf{A} est antisymétrique. On en déduit alors que :

$$\dot{\Delta}_k(t_1) = 0 \quad (2.49)$$

– **Remarque :** On peut retrouver très facilement le résultat $\dot{\Delta}_k(t_1) = 0$ pour une transformation orthogonale en remarquant que, dans ce cas, $\Delta_k = |\mathbf{R}| = 1, \forall t$.

En utilisant ce résultat, l'équation (2.44) peut s'écrire sous la forme :

$$\begin{aligned} \dot{m}_{p_1 \dots p_k} &= \sum_{i=1}^k \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} p_i x_i^{p_i-1} \dot{x}_i \prod_{j=1, j \neq i}^k x_j^{p_j} f(\mathbf{X}) d\mathbf{X} \\ &+ \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} x_1^{p_1} \dots x_k^{p_k} \dot{f}(\mathbf{X}) d\mathbf{X} \end{aligned} \quad (2.50)$$

Nous faisons à présent l'hypothèse que $\dot{f}(\mathbf{X}) = 0$ (c'est-à-dire que la fonction de densité ne change pas suite à la transformation \mathbf{R}). Cette hypothèse a été posée également dans [Reiss 91, Mamistvalov 98] pour démontrer le RFTMI. Sous cette hypothèse, le deuxième terme de (2.50) s'annule. En combinant (2.50) avec (2.34), on obtient finalement :

$$\dot{m}_{p_1 \dots p_k} = \sum_{i=1}^k \sum_{j=1, j \neq i}^k p_i a_{ij} m_{p_1 \dots p_{i-1} \dots p_j+1 \dots p_k} \quad (2.51)$$

où a_{ij} sont les éléments de la matrice \mathbf{A} (on rappelle que $\dot{\mathbf{X}}(t_1) = \mathbf{A}\mathbf{X}(t_1)$ d'après 2.34). À partir de (2.51), on peut noter que la dérivée de chaque élément de \mathbf{m}_p^1 est une combinaison linéaire des autres éléments de \mathbf{m}_p^1 . En utilisant (2.51), il est alors possible de calculer une certaine matrice $\mathbf{M}_1(\mathbf{A})$ de manière à ce que l'équation (2.36) s'écrive :

$$\dot{q}(\mathbf{m}_p^1) = \boldsymbol{\alpha}^T \mathbf{M}_1(\mathbf{A}) \mathbf{m}_p^1 = 0 \quad (2.52)$$

Le vecteur de paramètres $\boldsymbol{\alpha}$ doit donc appartenir au noyau de la matrice $\mathbf{M}_1^T(\mathbf{A})$ pour trouver une combinaison de moments invariante aux rotations.

Considérons maintenant le cas général où on cherche des polynômes invariants d'ordre n , soit de la forme :

$$q(\mathbf{m}_p^n) = \boldsymbol{\alpha}^T \mathbf{m}_p^n \quad (2.53)$$

où $\mathbf{m}_p^n = (m_{p\dots 0}^n, m_{p\dots 0}^{n-1}m_{p-1\dots 0}^1, \dots, m_{0\dots p}^n)$. Le polynôme $q(\mathbf{m}_p^n)$ est invariant à la transformation (2.33) si et seulement si :

$$\dot{q}(\mathbf{m}_p^n) = \boldsymbol{\alpha}^T \dot{\mathbf{m}}_p^n = 0, \forall \mathbf{m}_p^n \quad (2.54)$$

À partir de (2.51), on peut noter que la dérivée de chaque élément de \mathbf{m}_p^n est également une combinaison linéaire des autres éléments de \mathbf{m}_p^n . Il est donc possible de calculer une matrice $\mathbf{M}_n(\mathbf{A})$ telle que :

$$\dot{q}(\mathbf{m}_p^n) = \boldsymbol{\alpha}^T \mathbf{M}_n(\mathbf{A}) \mathbf{m}_p^n = 0 \quad (2.55)$$

Finalement, si $\boldsymbol{\alpha}$ appartient au noyau de $\mathbf{M}_n^T(\mathbf{A})$, alors $\boldsymbol{\alpha}^T \mathbf{m}_p^n$ est un invariant aux transformations orthogonales.

Exemples :

À partir de (2.51), on obtient $\dot{m}_{0\dots 0} = 0$. $m_{0\dots 0}$ est donc invariant aux rotations, ce qui était bien sûr attendu.

Dans le cas d'un espace de dimension $k = 2$, on a

$$\mathbf{A} = \begin{bmatrix} 0 & -w \\ w & 0 \end{bmatrix}$$

où w est la vitesse de rotation de l'objet dans son plan.

– Nous allons déterminer tout d'abord les polynômes invariants d'ordre 1 pour les moments d'ordre 2. Pour cela, on définit $\mathbf{m}_2^1 = [m_{20}, m_{11}, m_{02}]$. En utilisant (2.51), on obtient :

$$\begin{aligned} \dot{m}_{20} &= 2 \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1 \dot{x}_1 f(x_1, x_2) dx_1 dx_2 \\ &= -2w \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1 x_2 f(x_1, x_2) dx_1 dx_2 \\ &= -2w m_{11} \\ \dot{m}_{11} &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (\dot{x}_1 x_2 + x_1 \dot{x}_2) f(x_1, x_2) dx_1 dx_2 \\ &= w m_{20} - w m_{02} \\ \dot{m}_{02} &= 2w m_{11} \end{aligned}$$

Ce qui permet de déterminer $\mathbf{M}_1(\mathbf{A})$:

$$\mathbf{M}_1(\mathbf{A}) = \begin{bmatrix} 0 & -2w & 0 \\ w & 0 & -w \\ 0 & 2w & 0 \end{bmatrix}$$

À partir de cette matrice, on en déduit $\boldsymbol{\alpha} = (1, 0, 1)$. On retrouve ainsi le premier invariant de Hu, qui est le seul invariant d'ordre 1 que l'on peut obtenir à partir des moments d'ordre 2 (puisque $\dim \text{Ker } \mathbf{M}_1(\mathbf{A}) = 1$):

$$q(\mathbf{m}_2^1) = m_{20} + m_{02} \quad (2.56)$$

- Déterminons à présent les polynômes invariants d'ordre 2 pour les moments d'ordre 2. Pour cela, on définit :

$$\mathbf{m}_2^2 = (m_{20}^2, m_{20}m_{11}, m_{20}m_{02}, m_{11}^2, m_{11}m_{02}, m_{02}^2)$$

En utilisant la méthode proposée et après de simples développements, on obtient :

$$\mathbf{M}_2(\mathbf{A}) = \begin{bmatrix} 0 & -4\omega & 0 & 0 & 0 & 0 \\ \omega & 0 & -\omega & -2\omega & 0 & 0 \\ 0 & 2\omega & 0 & 0 & -2\omega & 0 \\ 0 & 2\omega & 0 & 0 & -2\omega & 0 \\ 0 & 0 & \omega & 2\omega & 0 & -\omega \\ 0 & 0 & 0 & 0 & 4\omega & 0 \end{bmatrix}$$

et donc

$$\text{Ker } \mathbf{M}_2^T(\mathbf{A}) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 2 \\ -1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.57)$$

On en déduit immédiatement deux invariants aux rotations :

$$q(\mathbf{m}_2^2) = m_{20}m_{02} - m_{11}^2 \quad (2.58)$$

$$q(\mathbf{m}_2^2) = m_{20}^2 + 2m_{20}m_{02} + m_{02}^2 \quad (2.59)$$

Ces invariants sont indépendants entre eux (puisque $\dim \text{Ker } \mathbf{M}_2(\mathbf{A}) = 2$). Cependant, (2.59) n'est rien d'autre que le carré de (2.56). L'utilisation de (2.59) est donc inutile. Dans le paragraphe suivant, nous verrons qu'il est facilement possible d'éliminer la redondance entre polynômes invariants. Par ailleurs, on peut noter que (2.58) peut être obtenu à partir des deux premiers invariants de Hu (donnés en (2.23)).

De nombreux autres invariants aux rotations peuvent être obtenus en utilisant la méthode proposée, en considérant soit des moments d'ordre supérieur, soit des espaces de dimension supérieure (par exemple $k = 3$). Une liste exhaustive des moments invariants aux rotations dans le cas de $k = 2$ et de $k = 3$ est donnée en annexe A pour les moments d'ordre inférieur à 6. Finalement, on rappelle que si les moments centrés sont utilisés dans les formules, les polynômes obtenus sont également invariants aux translations.

2.2.3.2 Élimination de la redondance

En asservissement visuel, quand le nombre d'informations visuelles est égal au nombre de degrés de liberté à commander, il est indispensable que les primitives utilisées soient indépendantes afin que la matrice d'interaction associée ne soit pas singulière. L'élimination de la redondance entre les invariants est également nécessaire dans d'autres applications comme l'indexation d'images. En effet, des invariants sont généralement utilisés pour la recherche d'images similaires à une image requête dans une base de données souvent de taille importante. Une grande dimension du vecteur requête peut être fatale autant au niveau temps de calcul qu'au niveau qualité des résultats de similarité entre l'image requête et les images de la base [Bellman 61, Verleysen 03, Donoho 00, Berrani 02]. Ce problème est connu dans la littérature sous le nom de la malédiction de la taille.

Dans le paragraphe précédent, on a déterminé le vecteur α qui intervient dans les polynômes invariants aux rotations à partir du noyau de la matrice $\mathbf{M}_n^T(\mathbf{A})$. Il existe donc un nombre infini de polynômes invariants de même ordre (par n'importe quelle combinaison linéaire des vecteurs de base de ce noyau). Par conséquent, il est nécessaire d'utiliser une base du noyau de la matrice $\mathbf{M}_n^T(\mathbf{A})$ pour éliminer la redondance entre les polynômes invariants du même ordre. Cependant, comme nous l'avons déjà vu pour (2.56) et (2.59), certains polynômes d'ordre n peuvent être obtenus à partir de polynômes d'ordres inférieurs. Pour éliminer cette dépendance, on peut utiliser la propriété suivante :

Propriété 1 : Soit \mathbf{E} l'espace associé au noyau de la matrice $\mathbf{M}_n^T(\mathbf{A})$. Soit \mathbf{E}_1 l'espace généré par les vecteurs α qui interviennent dans les polynômes invariants d'ordre n obtenus par un produit de polynômes d'ordre inférieur à n . Alors l'ensemble des polynômes d'ordre n mutuellement indépendants qui est également indépendant des polynômes d'ordre inférieur sont donnés par les vecteurs de la base de $\mathbf{E}_2 = \mathbf{E}/\mathbf{E}_1$

La propriété précédente peut être utilisée pour construire une base d'invariants pour les polynômes invariants obtenus dans le paragraphe précédent. Par exemple, les polynômes (2.56), (2.58) et (2.59) sont dépendants puisque comme on l'a déjà dit, le carré du polynôme (2.56) est un polynôme invariant d'ordre 2. Soit \mathbf{E}_1 l'espace des vecteurs définissant les polynômes d'ordre 2 pouvant être obtenus par produit de polynômes d'ordres 1. \mathbf{E}_1 est donc généré par le vecteur α_1 qui apparaît dans le carré du polynôme (2.56) :

$$\alpha_1 = [1, 0, 2, 0, 0, 1] \quad (2.60)$$

La base de l'espace \mathbf{E} des vecteurs qui apparaissent dans les polynômes invariants d'ordre 2 est donnée par (2.57), soit :

$$\alpha_2 = [1, 0, 2, 0, 0, 1], \alpha_3 = [0, 0, -1, 1, 0, 0] \quad (2.61)$$

On peut immédiatement en déduire que l'ensemble des polynômes invariants d'ordre deux indépendants des polynômes d'ordre 1 est donné par le vecteur α_3 , étant donné que $\alpha_1 = \alpha_2$.

2.2.3.3 Invariance à l'échelle

Dans le domaine de la reconnaissance d'objets, plusieurs applications utilisent la classification des objets selon leur forme (carrée, sphérique, cubique, etc). Ce genre d'applications nécessite des descripteurs invariants à l'échelle. Si on utilise les moments comme descripteurs, et si le changement d'échelle est donné par un facteur d'échelle uniforme, il est très facile de rendre les polynômes invariants aux translations et aux rotations également invariants aux changements d'échelle. En effet, si chaque composante de $\mathbf{X} = (x_1, \dots, x_k)$ est multipliée par un facteur λ , le polynôme $q(\mathbf{m}_p^n)$ sera multiplié par $\lambda^{n(p+k)}$ et $m_{0\dots 0}$ sera multiplié par λ^k . Le rapport entre $q(\mathbf{m}_p^n)$ et $m_{0\dots 0}^{n(\frac{p}{k}+1)}$ est donc invariant au changement uniforme d'échelle.

D'autres normalisations peuvent être envisagées pour obtenir des invariants au changement uniforme d'échelle. Par exemple, le rapport de deux moments du même ordre est invariant aux changements uniformes d'échelle, à condition que le dénominateur soit différent de zéro. Ainsi, on peut démontrer que le rapport de deux polynômes invariants du même ordre est invariant également aux changements d'échelle. Par exemple, le rapport :

$$r_1 = \frac{I_{n_1}}{I_{n_2}} = \frac{-\mu_{20}\mu_{02} + \mu_{11}^2}{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \quad (2.62)$$

est invariant aux rotations et au changement uniforme d'échelle. On peut également démontrer de la même façon que le rapport :

$$\frac{m_{p_1\dots p_k}}{(m_{2\dots 0} + \dots + m_{0\dots 2})^{\frac{p+2}{4}}}$$

est également un invariant aux changements uniformes d'échelle. On considère maintenant le cas général où le changement d'échelle est non uniforme. On rappelle que k est la dimension de l'espace considéré et p l'ordre des moments. Le cas général d'un changement d'échelle s'écrit :

$$\mathbf{X}' = \mathbf{D}\mathbf{X} \quad (2.63)$$

où $\mathbf{D} = \text{diag}(\lambda_1, \dots, \lambda_k)$. Mamistvalov dans [Mamistvalov 98] propose comme invariant au changement d'échelle (2.63) :

$$I = \frac{m_{\frac{p}{k}\dots\frac{p}{k}}}{m_{\frac{p}{k}\dots\frac{p}{k}}^{\frac{p}{k}+1}} \quad (2.64)$$

On propose une forme plus générale d'invariants aux changements d'échelle :

$$I = \frac{\prod_{i=1}^k m_{p_1^i \dots p_k^i}^{n_i}}{m_{0 \dots 0}^{n(\frac{p}{k}+1)}} \quad (2.65)$$

où $\sum_{i=1}^k (p_j^i + 1)n_i = n(\frac{p}{k} + 1) \forall j$, $\sum_{j=1}^k p_j^i = p \forall i$, et $\sum_{i=1}^k n_i = n$. En effet, en utilisant la définition des moments, on obtient après la transformation (2.63), $m_{p_1^i \dots p_k^i}^{n_i}$ multiplié par $\prod_{j=1}^k \lambda_j^{(p_j^i+1)n_i}$ et donc $\prod_{i=1}^k m_{p_1^i \dots p_k^i}^{n_i}$ multiplié par $\prod_{j=1}^k \lambda_j^{\sum_{i=1}^k (p_j^i+1)n_i}$. Puisque $m_{0 \dots 0}^{n(\frac{p}{k}+1)}$ est multiplié par $\prod_{j=1}^k \lambda_j^{n(\frac{p}{k}+1)}$, alors (2.65) est invariant à (2.63). En utilisant notre formule, on obtient comme exemple d'invariants à l'échelle :

$$\begin{aligned} I_1 &= \frac{m_{11}}{m_{00}^2}, & I_2 &= \frac{m_{02}m_{20}}{m_{00}^4}, & I_3 &= \frac{m_{30}m_{03}}{m_{00}^5}, \\ I_4 &= \frac{m_{21}m_{12}}{m_{00}^5}, & I_5 &= \frac{m_{22}}{m_{00}^3}, & I_6 &= \frac{m_{40}m_{04}}{m_{00}^6}, & I_7 &= \frac{m_{31}m_{13}}{m_{00}^6} \end{aligned} \quad (2.66)$$

On peut noter que les invariants à l'échelle proposés dans [Mamistvalov 98] sont un cas particulier de la forme générale (2.65) que nous proposons (c'est-à-dire valable pour $n = 1$). Ainsi, seuls les invariants I_1 et I_5 de (2.66) peuvent être obtenus en utilisant la formule (2.64) de Mamistvalov. De même, d'autres invariants peuvent être obtenus par addition, soustraction, ou multiplication de deux ou plusieurs invariants de la forme précédente. Par exemple, le rapport :

$$I = \frac{\prod_{i=1}^k m_{p_1^i \dots p_k^i}^{n_i}}{m_{2 \dots 2}^{n(\frac{p+k}{2(k+1)})}} \quad (2.67)$$

est également invariant aux changements d'échelle. Ainsi, on obtient comme exemples d'invariants aux changements d'échelle :

$$\begin{aligned} I'_1 &= \frac{m_{11}}{m_{22}^{\frac{2}{3}}}, & I'_2 &= \frac{m_{02}m_{20}}{m_{22}^{\frac{4}{3}}}, & I'_3 &= \frac{m_{30}m_{03}}{m_{22}^{\frac{5}{3}}}, \\ I'_4 &= \frac{m_{21}m_{12}}{m_{22}^{\frac{5}{3}}}, & I'_5 &= \frac{m_{40}m_{04}}{m_{22}^2}, & I'_6 &= \frac{m_{31}m_{13}}{m_{22}^2} \end{aligned} \quad (2.68)$$

2.2.3.4 Invariance aux rotations et à l'échelle

À partir des résultats obtenus précédemment, nous cherchons à présent des polynômes qui sont invariants aussi bien aux rotations qu'aux changements d'échelle. Dans le paragraphe 2.2.3.1, nous avons obtenu des polynômes invariants aux transformations orthogonales sous la forme suivante :

$$q(\mathbf{m}_p^n) = \boldsymbol{\alpha}^T \mathbf{m}_p^n \text{ où } \boldsymbol{\alpha} \in \text{Ker } \mathbf{M}_n^T(\mathbf{A})$$

Chaque élément du vecteur \mathbf{m}_p^n peut s'écrire sous la forme : $\prod_{i=1}^k m_{p_1^i \dots p_k^i}^{n_i}$ avec $\sum_{j=1}^k p_j^i = p$ et $\sum_{i=1}^k n_i = n$. On note α_i le $i^{\text{ème}}$ élément de $\boldsymbol{\alpha}$, et m_i le $i^{\text{ème}}$ élément de \mathbf{m}_p^n . Alors, si $\forall i$ tel que $\alpha_i \neq 0$, $\frac{m_p^n(s)}{m_{0 \dots 0}^{l(\frac{p}{k}+1)}}$ est invariant aux changements d'échelle, alors le rapport :

$$I_t = \frac{\boldsymbol{\alpha}^T \mathbf{m}_p^n}{m_{0 \dots 0}^{l(\frac{p}{k}+1)}} \quad (2.69)$$

est invariant aux transformations orthogonales et aux changements d'échelle. En effet, I_t est un rapport de deux invariants aux transformations orthogonales et une combinaison linéaire d'invariants aux changements d'échelle donnés par (2.65). Par exemple :

$$I_{t_1} = \frac{m_{20}m_{02} - m_{11}^2}{m_{00}^4} \quad (2.70)$$

et

$$I_{t_2} = \frac{m_{40}m_{04} - 4m_{31}m_{13} + 3m_{22}^2}{m_{00}^6} \quad (2.71)$$

sont invariants aux rotations et aux changements d'échelle (le numérateur de (2.71) étant invariant aux rotations). D'autres formules d'invariants peuvent étre obtenues en utilisant d'autres polynômes invariants au numérateur. Par exemple, le rapport :

$$I_{t_2} = \frac{m_{40}m_{04} - 4m_{31}m_{13} + 3m_{22}^2}{(m_{20}m_{02} - m_{11}^2)^2} \quad (2.72)$$

est invariant aux mouvements de rotation et aux changements d'échelle.

2.3 Généralisation au cas d'un ensemble de points

Dans cette section, on applique les méthodes développées dans la section précédente au cas de moments calculés à partir d'un ensemble de points, par exemple un ensemble de points d'intérêt extraits par l'algorithme du détecteur de Harris [Harris 88] dans le cas d'une image (voir figure 2.3).

2.3.1 Définitions

2.3.1.1 Moments 2D

- Les moments d'une distribution de N points sont définis par la relation classique :

$$m_{ij} = \sum_{h=0}^N x_h^i y_h^j \quad (2.73)$$

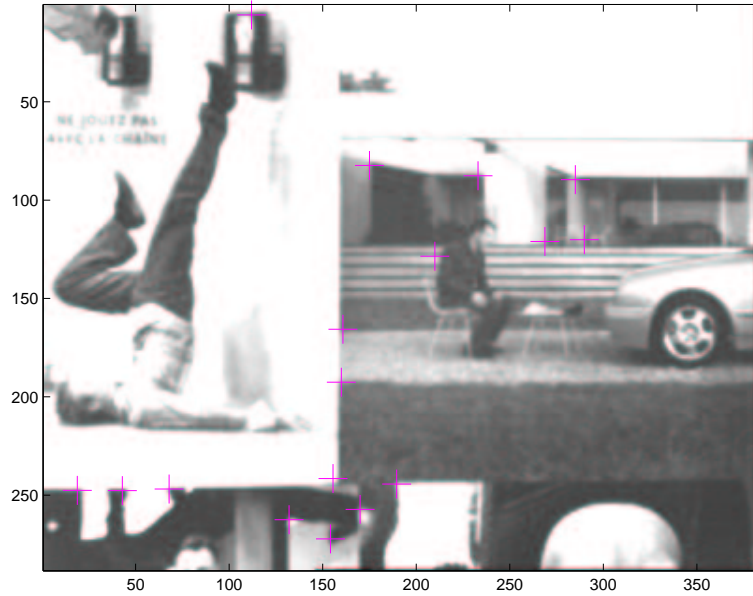


FIG. 2.3 – Exemple d'un ensemble de points extraits à l'aide du détecteur de Harris

- Les moments centrés μ_{pq} d'un ensemble de N points sont calculés comme dans le cas d'un objet continu par rapport au centre de gravité de l'objet (x_g, y_g) . Ils sont calculés par :

$$\mu_{ij} = \sum_{h=0}^N (x_h - x_g)^i (y_h - y_g)^j \quad (2.74)$$

où $x_g = \frac{m_{10}}{m_{00}}$ et $y_g = \frac{m_{01}}{m_{00}}$. Les moments centrés peuvent être calculés directement à partir des moments m_{pq} d'ordres inférieurs à $p + q + 1$ par l'équation (2.6).

2.3.1.2 Moments k -dimensionnels

- Les moments k -dimensionnels sont définis de la même manière que dans le cas d'un espace de dimension 2 par :

$$m_{p_1 \dots p_k} \equiv \sum_{h=0}^N x_{1,h}^{p_1} \dots x_{k,h}^{p_k} \quad (2.75)$$

$p = p_1 + \dots + p_k$ est l'ordre du moment $m_{p_1 \dots p_k}$. $x_{i,h}$ est la $i^{\text{ème}}$ composante de \mathbf{x}_h , où \mathbf{x}_h est le vecteur de coordonnées d'un des N points. Finalement, les moments k -dimensionnels centrés sont définis par :

$$\mu_{p_1 \dots p_k} \equiv \sum_{h=0}^N (x_{1,h} - x_{1g})^{p_1} \dots (x_{k,h} - x_{kg})^{p_k} \quad (2.76)$$

avec $x_{1g} = \frac{m_{10\dots 0}}{m_{0\dots 0}}$, \dots , $x_{kg} = \frac{m_{0\dots 01}}{m_{0\dots 0}}$. Ils peuvent être calculés à partir des moments donnés par l'équation (2.75) en utilisant l'équation (2.20).

Dans la suite, nous allons généraliser la méthode de détermination des invariants obtenue dans le cas d'un objet continu au cas d'un ensemble de points.

2.3.2 Théorème fondamental des moments invariants d'un ensemble de points

Théorème 2 Soit $H(x_1, \dots, x_k)$ une forme algébrique k – naire d'ordre p .
Si $H(x_1, \dots, x_k)$ a un invariant algébrique d'ordre n et de poids w :

$$I(a') = \Delta^w I(a) \quad (2.77)$$

alors les moments d'ordre p ont le même invariant et de même poids w :

$$I(m'_{p\dots 0} \dots m'_{0\dots p}) = \Delta^w I(m_{p\dots 0} \dots m_{0\dots p}) \quad (2.78)$$

Ce théorème peut être démontré de la même façon que dans le cas continu. La seule différence entre les deux cas est le facteur additif Δ^n dans le cas continu, qui est dû à la variation de l'élément $dx_1 \dots dx_k$ dans l'intégrale multiple après une transformation affine.

2.3.3 Invariants aux rotations

Comme dans le cas continu, on cherche des invariants aux transformations orthogonales sous forme polynomiale d'ordre n :

$$q(\mathbf{m}_p^n) = \boldsymbol{\alpha}^T \mathbf{m}_p^n \quad (2.79)$$

Le polynôme $q(\mathbf{m}_p^n)$ est donc invariant à (2.33) si et seulement si :

$$\dot{q}(\mathbf{m}_p^1) = \boldsymbol{\alpha}^T \dot{\mathbf{m}}_p^1 = 0, \quad \forall \mathbf{m}_p^1 \quad (2.80)$$

Après dérivation de (2.75), on obtient :

$$\dot{m}_{p_1 \dots p_k} = \sum_{i=1}^k \sum_{h=0}^N p_i x_i^{p_i-1} \dot{x}_i \prod_{j=1, j \neq i}^k x_j^{p_j} \quad (2.81)$$

En combinant (2.81) avec (2.34), on obtient :

$$\dot{m}_{p_1 \dots p_k} = \sum_{i=1}^k \sum_{j=1, j \neq i}^k p_i a_{ij} m_{p_1 \dots p_{i-1} \dots p_j+1 \dots p_k} \quad (2.82)$$

On rappelle que les a_{ij} sont les éléments de la matrice $\mathbf{A} = \mathbf{R}^T \dot{\mathbf{R}}$. On note alors comme dans le cas continu que la dérivée de $m_{p_1 \dots p_k}$ est une combinaison linéaire des moments du même ordre. Il est donc possible de calculer une matrice $\mathbf{M}_n(\mathbf{A})$ telle que :

$$\dot{q}(\mathbf{m}_p^n) = \boldsymbol{\alpha}^T \mathbf{M}_n(\mathbf{A}) \mathbf{m}_p^n = 0 \quad (2.83)$$

En comparant les équations (2.82) et (2.51), on note que la matrice $\mathbf{M}_n(\mathbf{A})$ a exactement la même expression dans les deux cas. Les invariants aux transformations orthogonales obtenus dans le cas continu sont donc aussi des invariants aux transformations orthogonales dans le cas des moments calculés à partir de N points. Si $\boldsymbol{\alpha}$ appartient au noyau de $\mathbf{M}_n^T(\mathbf{A})$, alors $\boldsymbol{\alpha}^T \mathbf{m}_p^n$ est un invariant aux transformations orthogonales.

2.3.4 Invariants aux changements d'échelle

Les invariants aux changements d'échelle proposés dans le paragraphe 2.2.3.3 dans le cas d'un objet continu ne sont plus valables dans le cas des moments calculés à partir d'un ensemble de points. En effet, le moment d'ordre 0 (c'est-à-dire $m_{0 \dots 0} = N$) est une constante quelque soit le changement d'échelle considéré). Cependant, plusieurs formes générales d'invariants aux changements d'échelle peuvent être proposées. En effet, si on considère un changement d'échelle uniforme, le rapport $\frac{m_{p_1 \dots p_k}}{(m_{2 \dots 0} + \dots + m_{0 \dots 2})^{\frac{p}{2}}}$ peut être utilisé comme forme générale d'invariants. En effet, en utilisant (2.73), on a après un changement d'échelle par un scalaire λ :

$$m'_{p_1 \dots p_k} = \lambda^p m_{p_1 \dots p_k} \quad (2.84)$$

On en déduit que :

$$(m'_{2 \dots 0} + \dots + m'_{0 \dots 2})^{\frac{p}{2}} = \lambda^p (m_{2 \dots 0} + \dots + m_{0 \dots 2})^{\frac{p}{2}} \quad (2.85)$$

En combinant les deux équations précédentes, on obtient facilement que le rapport $\frac{m_{p_1 \dots p_k}}{(m_{2 \dots 0} + \dots + m_{0 \dots 2})^{\frac{p}{2}}}$ est bien invariant au changements uniformes d'échelle.

Dans le cas général de changement d'échelle donné par (2.63), on peut montrer de la même façon que dans le cas continu, que le rapport donné ci-dessous est invariant aux changements d'échelle :

$$I = \frac{\prod_{i=1}^k m_{p_1^i \dots p_k^i}^{n_i}}{m_{2 \dots 2}^{n(\frac{p}{2k})}} \quad (2.86)$$

où $\sum_{i=1}^k p_j^i n_i = \frac{np}{k} \forall j$, $\sum_{j=1}^k p_j^i = p \forall i$, et $\sum_{i=1}^k n_i = n$. Voici des exemples d'invariants pouvant être obtenus en utilisant (2.86) dans le cas 2D :

$$\begin{aligned} I_1 &= \frac{m_{11}}{m_{22}^{\frac{1}{2}}}, & I_2 &= \frac{m_{02}m_{20}}{m_{22}}, & I_3 &= \frac{m_{30}m_{03}}{m_{22}^{\frac{3}{2}}}, \\ I_4 &= \frac{m_{21}m_{12}}{m_{22}^{\frac{3}{2}}}, & I_5 &= \frac{m_{40}m_{04}}{m_{22}^2}, & I_6 &= \frac{m_{31}m_{13}}{m_{22}^2} \end{aligned} \quad (2.87)$$

Finalement, si les moments sont calculés par rapport au centre de gravité du nuage de points (c'est-à-dire si on utilise les moments centrés), alors ils sont également invariants aux mouvements de translations. On peut de même combiner les invariants aux rotations, aux translations et aux changements d'échelle en utilisant la même méthode proposée dans le cas continu (voir paragraphe 2.2.3.4).

2.4 Conclusion

Dans ce chapitre, nous avons développé une nouvelle méthode de calcul des moments invariants (invariants aux translations, aux rotations et aux changements d'échelle). La méthode proposée est systématique et valable dans un espace de dimension quelconque. Elle est également valable dans le cas des moments d'un objet continu et dans le cas des moments calculés à partir d'un nuage de points. À partir de cette méthode, on a établi une règle simple pour éliminer la redondance entre les invariants obtenus. Les moments invariants vont être utilisés dans la suite de ce manuscrit pour obtenir six informations visuelles indépendantes pour contrôler les six degrés de liberté de la caméra.

Chapitre 3

Asservissement visuel à partir des moments

3.1 Introduction

Les techniques d'asservissement visuel consistent à utiliser les informations fournies par une ou plusieurs caméras afin de contrôler les mouvements d'un système robotique. On peut ainsi réaliser une grande variété de tâches de positionnement du système par rapport à son environnement, ou de poursuite d'objets mobiles, en contrôlant de un à l'ensemble des m degrés de liberté du système. On distingue deux configurations du capteur de vision par rapport au robot :

- Caméra embarquée : c'est la configuration la plus répandue. La caméra est fixée rigidement à l'organe terminal du manipulateur de manière à ce que les objets situés dans l'espace du travail soient dans son champ de vision.
- Caméra(s) déportée(s) : dans cette configuration, le système de vision est positionné de manière à ce que l'organe terminal du robot et les objets situés dans l'espace du travail soient dans son champ de vision. La caméra peut être fixe ou montée sur un autre système mécanique, une tourelle pan-tilt par exemple.

Quelle que soit la configuration du capteur, il s'agit de sélectionner au mieux un ensemble de k mesures, permettant de contrôler les m degrés de liberté souhaités, et d'élaborer une loi de commande afin que ces mesures $\mathbf{s}(t)$ atteignent une valeur désirée \mathbf{s}^* , définissant une réalisation correcte de la tâche. On peut également vouloir suivre une trajectoire $\mathbf{s}^*(t)$. Le principe de la commande se ramène donc à réguler (c'est-à-dire amener et maintenir à zéro) le vecteur d'erreur $\mathbf{s}(t) - \mathbf{s}^*(t)$. Dans la suite de ce manuscrit, nous nous intéressons essentiellement au cas où \mathbf{s}^* est constante. La réalisation de ce genre de tâches nécessite l'estimation ou l'approximation de la relation qui lie la variation des mesures choisies $\mathbf{s}(t)$ aux variables de contrôle. Dans la suite de ce chapitre, nous traitons respectivement les points suivants :

- La commande par asservissement visuel et la notion de matrice d'interaction.
- Les problèmes du couplage de la commande et des non linéarités présentes dans la matrice d'interaction.
- L'obtention des formules analytiques des matrices d'interaction associées aux moments calculés à partir d'une projection plane et d'une projection sphérique. Les cas où les moments sont calculés à partir d'une distribution continue et à partir d'un ensemble discret de points sont considérés.
- Le choix des informations visuelles à partir des moments pour résoudre les problèmes du couplage de la commande et des non linéarités de la matrice d'interaction.

3.2 Matrice d'interaction et loi de commande

Dans ce paragraphe, nous rappelons quelques notions sur la commande par asservissement visuel. Plus de détails sur ces notions sont donnés dans [Chaumette 02a]. Un ensemble \mathbf{s} de k informations visuelles peut être pris en compte dans un schéma d'asservissement visuel, à partir du moment où il est défini par la donnée d'une application différentiable de SE_3 dans \mathbb{R}^k :

$$\mathbf{s} = \mathbf{s}(\mathbf{r}(t)) \quad (3.1)$$

où $\mathbf{r}(t)$, élément de l'espace SE_3 de représentation des repères et des corps rigides, décrit la situation à l'instant t entre la caméra et son environnement. On considère donc que seuls des mouvements de la caméra ou des objets qu'elle perçoit sont susceptibles de faire varier la valeur d'une information visuelle. La différentielle de \mathbf{s} permet de relier les variations des informations visuelles au mouvement relatif entre la caméra et la scène. On obtient en effet en dérivant (3.1) :

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \dot{\mathbf{r}} = \mathbf{L}_s \mathbf{v} \quad (3.2)$$

où :

- \mathbf{L}_s est une matrice de type jacobienne de dimension $k \times 6$, appelée la **matrice d'interaction** associée à \mathbf{s} . La valeur de cette matrice dépend de \mathbf{s} et également de la position relative entre l'objet et la caméra.
- \mathbf{v} est le torseur cinématique relatif entre la caméra et la scène, exprimé dans le repère de la caméra R_c en son origine C . Plus précisément, si \mathbf{v}_c et \mathbf{v}_o sont respectivement les torseurs cinématiques de la caméra et de la scène qu'elle perçoit, tous deux exprimés dans R_c et en C , on a :

$$\mathbf{v} = \mathbf{v}_c - \mathbf{v}_o \quad (3.3)$$

On note par ailleurs v la vitesse de translation à l'origine du repère et ω la vitesse de rotation telles que $\mathbf{v} = (v, \omega)$. Si ${}^o\mathbf{R}_c$ décrit la matrice de rotation

entre un repère R_o lié à l'objet et R_c , on a par définition [Samson 91]:

$$[\omega]_{\times} = {}^o\dot{\mathbf{R}}_c {}^o\mathbf{R}_c^T = -{}^c\dot{\mathbf{R}}_o {}^c\mathbf{R}_o^T = {}^c\mathbf{R}_o {}^o\dot{\mathbf{R}}_c \quad (3.4)$$

où $[\omega]_{\times}$ est la matrice antisymétrique du produit vectoriel associé à ω .

Dans la littérature, de nombreux types de lois de commande ont été proposés: loi de commande non linéaire [Hashimoto 93, Reyes 98], optimale de type LQ ou LQG [Papanikolopoulos 93, Hashimoto 96], basée sur un contrôleur GPC [Gangloff 98, Gangloff 99], voire robuste H_{∞} [Khadraoui 96] ou par retour d'état continu instationnaire dans le cas de robots mobiles présentant des contraintes non-holonomes [Tsakiris 98]. Dans ce manuscrit, nous allons simplement nous intéresser à réaliser une décroissance exponentielle de la fonction de tâche \mathbf{e} , définie par:

$$\mathbf{e}(\mathbf{r}(t)) = \mathbf{C} (\mathbf{s}(\mathbf{r}(t)) - \mathbf{s}^*) \quad (3.5)$$

où:

- $\mathbf{s}(\mathbf{r}(t))$ est la valeur courante des informations visuelles sélectionnées;
- \mathbf{s}^* est la valeur que \mathbf{s} doit atteindre pour réaliser la tâche;
- \mathbf{C} est une matrice $6 \times k$ de rang plein 6, dite de combinaison, telles que les 6 composantes de \mathbf{e} soient indépendantes et contrôlent les 6 degrés de liberté du système.

En utilisant (3.5) et (3.2), si la matrice \mathbf{C} est choisie constante, la différentielle de \mathbf{e} est donnée par:

$$\dot{\mathbf{e}} = \mathbf{C} \dot{\mathbf{s}} = \mathbf{C} \mathbf{L}_s \mathbf{v} \quad (3.6)$$

Si on désire obtenir une décroissance exponentielle de la fonction de tâche, on a alors:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (3.7)$$

où λ est un gain scalaire régissant la vitesse de décroissance de \mathbf{e} . En supposant que la caméra est embarquée et que l'objet est fixe ($\mathbf{v}_o = 0$), on obtient:

$$\mathbf{v}_c = -\lambda \left(\mathbf{C} \widehat{\mathbf{L}}_s \right)^{-1} \mathbf{e} \quad (3.8)$$

où $\widehat{\mathbf{L}}_s$ est une estimation de la valeur de la matrice d'interaction. En combinant (3.6) et (3.8), on obtient:

$$\dot{\mathbf{e}} = -\lambda \mathbf{C} \mathbf{L}_s \left(\mathbf{C} \widehat{\mathbf{L}}_s \right)^{-1} \mathbf{e} \quad (3.9)$$

La condition de positivité:

$$\mathbf{C} \mathbf{L}_s \left(\mathbf{C} \widehat{\mathbf{L}}_s \right)^{-1} > 0 \quad (3.10)$$

suffit donc pour assurer la décroissance de $\|\mathbf{e}\|$. Nous discutons dans la suite, les choix possibles pour la matrice \mathbf{C} . Nous distinguons deux cas de figures ($k > 6$) et ($k = 6$).

3.2.1 Cas où \mathbf{s} est de dimension supérieure à 6 ($k > 6$)

Dans ce cas, le choix le plus simple consiste à choisir pour \mathbf{C} la pseudo-inverse d'une approximation de la valeur de la matrice d'interaction à la position désirée :

$$\mathbf{C} = \widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ \quad (3.11)$$

Le calcul de \mathbf{C} nécessite donc de connaître la valeur de \mathbf{s}^* et des paramètres de profondeur à la position désirée. Si le modèle 3D de l'objet est disponible, ceux-ci peuvent se calculer aisément par un calcul de pose en utilisant l'image désirée. Autrement, ils sont en général fixés lors de la spécification même de la tâche. En utilisant (3.11), la loi de commande (3.8) s'écrit :

$$\mathbf{v}_c = -\lambda \left(\widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ \widehat{\mathbf{L}}_{\mathbf{s}} \right)^{-1} \mathbf{e} \quad (3.12)$$

et, en choisissant $\widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}$ pour approximer $\widehat{\mathbf{L}}_{\mathbf{s}}$, on obtient finalement :

$$\mathbf{v}_c = -\lambda \mathbf{e} = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ (\mathbf{s} - \mathbf{s}^*) \quad (3.13)$$

La condition de stabilité (3.10) s'écrit à présent :

$$\widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+ \mathbf{L}_{\mathbf{s}} > 0 \quad (3.14)$$

qui n'est vérifiée a priori qu'autour de la position désirée. Par ailleurs, une annulation de l'erreur (c'est-à-dire $\mathbf{e} = 0$) n'implique pas forcément une annulation des erreurs sur les primitives (c'est-à-dire $\mathbf{s} = \mathbf{s}^*$). En effet, l'ensemble des configurations telles que :

$$(\mathbf{s} - \mathbf{s}^*) \in \text{Ker } \mathbf{C} \quad (3.15)$$

correspond à un minimum local de la commande.

Un autre choix de \mathbf{C} qui tente de remédier aux problèmes précédents est de prendre directement la pseudo-inverse de la matrice d'interaction à chaque itération :

$$\mathbf{C} = \widehat{\mathbf{L}}_{\mathbf{s}}^+ \quad (3.16)$$

ce qui aboutit à la loi de commande :

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}}^+ (\mathbf{s} - \mathbf{s}^*) \quad (3.17)$$

Dans ce cas, un calcul de pose est nécessaire à chaque itération pour estimer les paramètres de profondeur qui interviennent dans la matrice d'interaction. L'avantage de cette méthode est qu'en cas de succès, elle fournit des bonnes trajectoires des primitives dans l'image (typiquement des droites si l'on considère des coordonnées de points dans \mathbf{s}). Cependant, en pratique, le comportement du système n'est pas

toujours satisfaisant puisque le mouvement du robot pour assurer ces trajectoires dans l'image peut être parfois étonnant. En plus, la condition de stabilité donnée par (3.10) ne s'applique plus puisque la matrice \mathbf{C} n'est pas constante. Par ailleurs, le problème des minima locaux lié au fait que le nombre de primitives est supérieur au nombre de degrés de liberté ($(\mathbf{s} - \mathbf{s}^*) \in \text{Ker } \mathbf{C}$) est tout aussi présent, l'existence de ces minima dans l'espace de travail étant démontrée dans [Chaumette 98].

Dans le paragraphe suivant, nous décrivons le cas où \mathbf{s} est de dimension égale à 6 ($k = 6$).

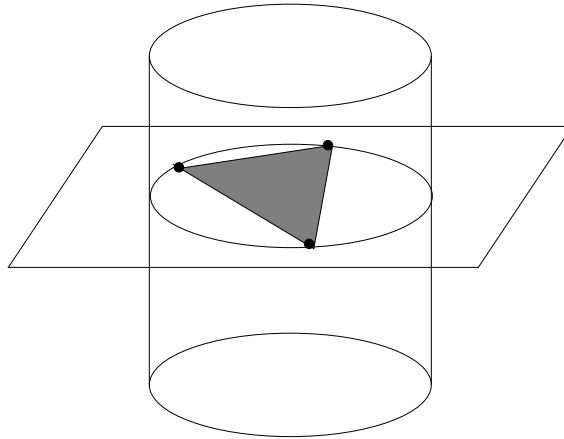
3.2.2 Cas où \mathbf{s} est de dimension égale à 6 ($k = 6$)

Pour commander les 6 degrés de liberté de la caméra, et si l'on dispose de 6 informations visuelles indépendantes pour cela, on a tout intérêt à choisir $\mathbf{C} = \mathbf{I}_6$. En utilisant de nouveau (3.7), la loi de commande à utiliser pour une caméra embarquée observant un objet statique est alors donnée par :

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_s^{-1} \mathbf{e} = -\lambda \widehat{\mathbf{L}}_s^{-1} (\mathbf{s} - \mathbf{s}^*) \quad (3.18)$$

Le choix d'utiliser la valeur courante de la matrice d'interaction dans la boucle de commande permet d'avoir un comportement idéal des primitives dans l'image. Cependant, ce choix peut générer des mouvements de la caméra inadéquats. Il faut aussi veiller à ce que la matrice d'interaction soit de rang plein tout au long de la trajectoire à réaliser. Considérons par exemple le cas où \mathbf{s} est constitué d'information visuelles 2D, comme les coordonnées de la projection de trois points dans l'image. La matrice d'interaction associée \mathbf{L}_s est alors de dimension 6×6 et est, dans la plupart des cas, de rang plein 6. Mais, on peut montrer [Michel 93, Papanikolopoulos 95] que certaines configurations entraînent une perte de rang de \mathbf{L}_s . En l'occurrence, ces singularités sont telles que les trois points soient alignés dans l'image ou que le centre optique C de la caméra appartienne à la surface du cylindre défini par le cercle circonscrit aux trois points (voir figure 3.1). Il est donc difficile d'assurer que, quelle que soit la position initiale choisie, le mouvement du robot évitera de traverser une singularité isolée.

De plus, il existe généralement quatre situations distinctes entre la caméra et la scène telles que l'image de trois points soit la même [Dhome 89]. La minimisation de $\|\mathbf{s} - \mathbf{s}^*\|$ peut donc amener le robot à l'un des quatre minima globaux tels que $\|\mathbf{s} - \mathbf{s}^*\| = 0$. Ainsi, dans ce cas, on peut très bien avoir $\mathbf{s} - \mathbf{s}^* = 0$ sans que la situation atteinte soit celle qui a été spécifiée. En considérant des informations visuelles de nature différente (par exemple les 3 droites que l'on peut définir à partir de 3 points non alignés), on retombe bien évidemment sur les mêmes problèmes potentiels. Pour y remédier, une solution pratique consiste à restreindre l'espace de travail à des zones proches de la situation désirée, ne comportant pas de singularités isolées, et où la minimisation de $\|\mathbf{s} - \mathbf{s}^*\|$ attire l'effecteur du robot vers sa position désirée. La détermination de la taille de ces zones est cependant très problématique. Il est également possible d'utiliser six informations visuelles 3D ou

FIG. 3.1 – *Cylindre de singularité*

2D1/2 pour contrôler les six degrés de liberté du robot. Dans ce cas, la matrice d'interaction est généralement de rang plein. Cependant, l'utilisation de ce genre d'information présente plusieurs problèmes mentionnés dans le premier chapitre de ce manuscrit.

Enfin, un autre choix consiste à utiliser la valeur de la matrice d'interaction estimée pour la position désirée. Dans ce cas, le problème de la singularité de la matrice ne se pose pas. Cependant, la condition de stabilité n'est vérifiée qu'autour de la position désirée de la caméra, avec des possibles passages de l'objet hors du champ de vue de la caméra.

Finalement, nous utiliserons la valeur moyenne des matrices d'interaction calculées pour les positions désirée et courante de la caméra dans la loi de commande :

$$\widehat{\mathbf{L}}_{\mathbf{s}} = \frac{1}{2}(\widehat{\mathbf{L}}_{\mathbf{s}} + \widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}) \quad (3.19)$$

Ce choix donne un bon comportement des informations visuelles et des vitesses de la caméra [Tahri 03], comme cela a été tout récemment démontré dans [Malis 04]

Dans la suite, nous discutons l'effet du choix des primitives sur le comportement du système.

3.2.3 Choix des informations visuelles

Plusieurs types de primitives \mathbf{s} ont été proposés auparavant. La plupart des travaux antérieurs ont considéré des objets simples qui peuvent être exprimés par des primitives géométriques de type points, droites, ellipses, etc. Par conséquent, la famille des objets pouvant être traités par ce type de méthodes est limitée. Par ailleurs, ce genre de méthodes nécessite la mise en correspondance entre les primitives dans un couple d'images, ce qui n'est pas toujours facile à obtenir. D'autres méthodes ont été proposées pour surmonter les problèmes mentionnés auparavant en utilisant par exemple la projection dans l'espace propre [Deguchi 00] ou la signature polaire

des points contours de l'objet [Collewet 00]. Récemment, une nouvelle méthode utilisant les moments dans l'image a été proposée dans [Chaumette 02b]. Dans la suite, nous proposons une large amélioration de cette méthode.

Nous essayons en effet de déterminer quelles sont les primitives qui permettent d'avoir un comportement optimal du système. La première condition nécessaire à assurer est que la matrice d'interaction soit non singulière. Un bon choix de primitives doit donc permettre d'obtenir une matrice d'interaction \mathbf{L}_s de rang plein 6 dans un domaine le plus large possible. Un bon moyen pour assurer cette condition est de réaliser un schéma de commande découplé, c'est-à-dire, pour chaque degré de liberté à commander, pouvoir extraire de l'image une caractéristique qui ne dépende que de ce degré de liberté. Dans ce cas de figure, la matrice d'interaction associée aux informations visuelles serait diagonale. Malheureusement, ce découplage est idéal et certainement impossible à atteindre. Sans être aussi exigeant, nous pourrions nous contenter d'une matrice diagonale par bloc ou triangulaire par bloc qui découplerait les rotations des translations. Un tel découplage permettra de déterminer et d'éviter facilement les singularités. En pratique, il peut être obtenu en utilisant un asservissement 3D [Wilson 96, Martinet 96], mais cela nécessite une connaissance du modèle 3D de l'objet. Ce découplage peut être obtenu également en utilisant un asservissement visuel $2D\ 1/2$ [Malis 99], où la connaissance du modèle de l'objet n'est pas nécessaire. Par contre une matrice d'homographie doit être déterminée à chaque itération (ce qui rend cette méthode légèrement plus sensible aux erreurs de mesure dans l'image, notamment aux abords de la convergence). En asservissement 2D, de premières tentatives ont été proposées récemment dans [Corke 01, Chaumette 02b]. Dans ce chapitre, une méthode plus efficace basée sur les moments invariants est proposée.

En asservissement visuel 2D, le comportement des primitives dans l'image est généralement satisfaisant. Cependant, la trajectoire du robot est imprévisible et peu satisfaisante pour de grands déplacements en rotation [Chaumette 98]. En fait, la différence entre le comportement dans l'image et l'espace 3D est due au non-linéarités dans la matrice d'interaction. Dans la suite, nous allons mettre en évidence ce problème à travers les matrices d'interaction de primitives usuelles en asservissement visuel.

3.3 Matrice d'interaction de primitives usuelles

3.3.1 Matrice d'interaction d'un point

Les algorithmes de traitement d'images fournissent des données exprimées en pixels. En négligeant les phénomènes fortement non linéaires de distorsion dus par exemple à l'emploi d'objectifs à très courte focale, le changement de variable pour passer des coordonnées (x_p, y_p) d'un point, exprimées en pixels, aux coordonnées

$\mathbf{x} = (x, y)$ de ce même point, mais exprimées en mètre, est donné par :

$$x = (x_p - x_c)/f_x \quad , \quad y = (y_p - y_c)/f_y \quad (3.20)$$

où (x_c, y_c) représente les coordonnées dans l'image du point principal et où $f_x = f/l_x$ et $f_y = f/l_y$ sont les rapports entre la distance focale f de l'objectif et les dimensions l_x et l_y d'un pixel [Horaud 95]. Ces paramètres, appelés paramètres intrinsèques de la caméra, peuvent être estimés au préalable lors d'une phase de calibration [Beyer 92, Robert 3]. Dans la suite de ce manuscrit, nous supposons que les coordonnées des points dans l'image sont exprimées en mètres. De manière très classique, le modèle mathématique d'une caméra est défini alors par une projection perspective telle que tout point M ayant pour coordonnées $\mathbf{X} = (X, Y, Z)$ se projette sur le plan image en un point m de coordonnées $\mathbf{x} = (x, y)$ avec :

$$x = X/Z \quad , \quad y = Y/Z \quad (3.21)$$

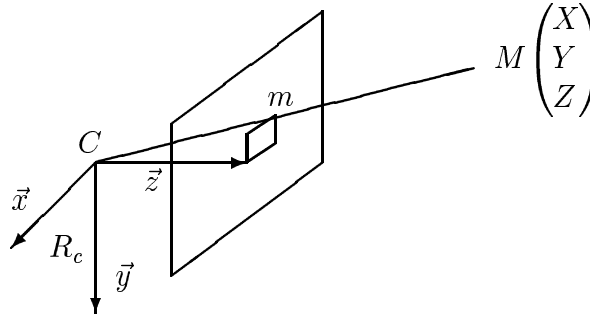


FIG. 3.2 – *Modèle de la caméra*

En différenciant cette équation, on obtient les variations dans l'image des coordonnées \mathbf{x} de m par rapport à la vitesse $\dot{\mathbf{X}}$ des coordonnées du point M :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 1/Z & 0 & -X/Z^2 \\ 0 & 1/Z & -Y/Z^2 \end{pmatrix} \dot{\mathbf{X}} \quad (3.22)$$

Quelle que soit la configuration choisie (caméra embarquée ou déportée, point M fixe ou mobile), la vitesse $\dot{\mathbf{X}}$ de M en fonction du torseur cinématique \mathbf{v} entre la caméra et son environnement est donnée par l'équation fondamentale de la cinématique :

$$\dot{\mathbf{X}} = -\mathbf{v} - \boldsymbol{\omega} \times \mathbf{X} = -\mathbf{v} + [\mathbf{X}]_{\times} \boldsymbol{\omega} = \begin{pmatrix} -\mathbb{I}_3 & [\mathbf{X}]_{\times} \end{pmatrix} \mathbf{v} \quad (3.23)$$

L'équation (3.22) peut alors se simplifier, en utilisant l'équation (3.21), sous la forme :

$$\dot{\mathbf{x}} = \mathbf{L}_{\mathbf{x}}(\mathbf{x}, Z) \mathbf{v} \quad (3.24)$$

où :

$$\mathbf{L}_{\mathbf{x}}(\mathbf{x}, Z) = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \quad (3.25)$$

On peut noter que les termes induits par les mouvements de rotation ne dépendent que des mesures x et y dans l'image. Par contre, les termes induits par les mouvements de translation sont inversement proportionnels à la profondeur du point 3D considéré. Ce phénomène se retrouve pour toutes les informations visuelles qu'il est possible de définir dans l'image (et décrit l'ambiguïté classique en vision par ordinateur entre l'amplitude d'un mouvement de translation et la profondeur des objets). En asservissement visuel, il faudra donc injecter une information 3D, pourtant a priori inconnue, dès que l'on souhaitera contrôler des degrés de liberté du robot impliquant des mouvements de translation. On peut également remarquer que la dynamique de \dot{x} et \dot{y} par rapport à chaque composante de la vitesse de la caméra n'est pas la même : quelques-unes sont inversement proportionnelles à la profondeur Z d'un point, certaines sont proportionnelles aux coordonnées de l'image du point et d'autres sont dépendantes du carré de ces coordonnées. Même s'il est possible de réaliser un schéma de commande tel que l'on obtienne une décroissance exponentielle découplée de l'erreur dans l'image, la dynamique du robot quant à elle sera très éloignée d'une décroissance exponentielle, du fait de la forte non linéarité de la matrice d'interaction $\mathbf{L}_{\mathbf{x}}$. La trajectoire du robot sera donc loin d'une trajectoire optimale.

Un problème important reste donc à résoudre : déterminer des primitives dont la matrice d'interaction associée est la moins non linéaire possible.

3.3.2 Matrice d'interaction d'une primitive géométrique paramétrable : cas d'une droite

On peut également calculer la matrice d'interaction associée à des informations visuelles construites à partir de primitives géométriques paramétrables [Chauvette 90, Espiau 92]. Pour cela, il suffit de définir les équations représentant :

- la nature et la configuration de la primitive dans la scène :

$$\mathbf{h}(\mathbf{X}, \mathbf{P}) = 0 \quad (3.26)$$

- sa projection dans le plan image :

$$\mathbf{g}(\mathbf{x}, \mathbf{p}) = 0 \quad (3.27)$$

- le lien entre la primitive 3D et son image (appelée dans le cas d'une primitive volumétrique la surface des limbes, voir figure 3.3) :

$$1/Z = \mu(\mathbf{x}, \mathbf{P}) = 0, \quad (3.28)$$

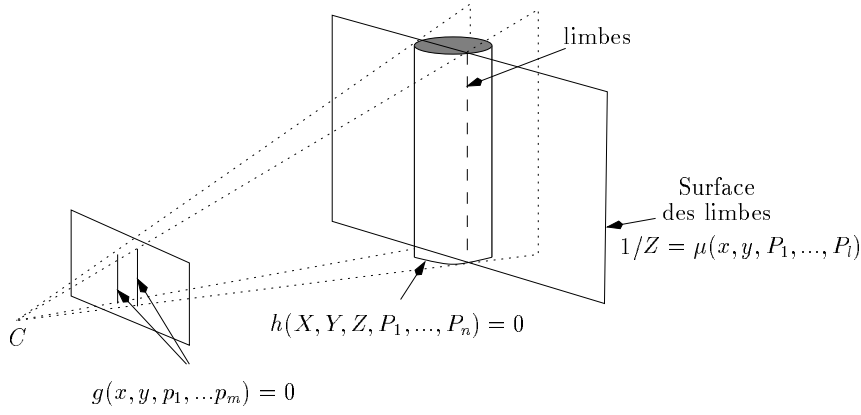


FIG. 3.3 – Projection de la primitive dans l'image et surface des limbes dans le cas d'un cylindre

A titre d'exemple, si une droite de l'espace est représentée par l'intersection des deux plans suivants :

$$\mathbf{h}(\mathbf{X}, \mathbf{P}) = \begin{cases} h_1 = A_1X + B_1Y + C_1Z + D_1 = 0 \\ h_2 = A_2X + B_2Y + C_2Z = 0 \end{cases} \quad (3.29)$$

on obtient immédiatement en utilisant les équations de la projection perspective (3.21) :

– la fonction μ à partir de h_1 :

$$1/Z = \alpha x + \beta y + \gamma \quad (3.30)$$

avec $\alpha = -A_1/D_1$, $\beta = -B_1/D_1$ et $\gamma = -C_1/D_1$.

– l'équation de la droite 2D, notée \mathcal{D} , résultant de la projection dans l'image de la droite 3D, à partir de h_2 :

$$ax + by + c = 0 \text{ avec } a = A_2, b = B_2, c = C_2 \quad (3.31)$$

La paramétrisation (a, b, c) n'étant pas minimale, on préfère choisir la représentation $\mathbf{p} = (\rho, \theta)$ définie par :

$$g(\mathbf{x}, \mathbf{p}) = x \cos \theta + y \sin \theta - \rho = 0 \quad (3.32)$$

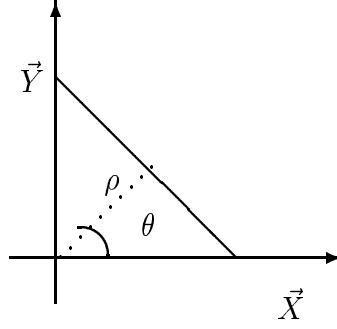
où $\theta = \arctan(b/a)$ et $\rho = -c/\sqrt{a^2 + b^2}$ (voir figure 3.4).

En différenciant l'équation (3.32), ce qui correspond à l'hypothèse que l'image d'une droite restera une droite quels que soient les mouvements de la caméra, on obtient :

$$\dot{\rho} + (x \sin \theta - y \cos \theta) \dot{\theta} = \dot{x} \cos \theta + \dot{y} \sin \theta, \quad \forall (x, y) \in \mathcal{D} \quad (3.33)$$

A partir de l'équation (3.32), on écrit x en fonction de y si $\cos \theta \neq 0$ (ou y en fonction de x sinon) et l'équation (3.33) peut alors s'écrire, en utilisant (3.24) et (3.30) :

$$(\dot{\rho} + \rho \tan \theta \dot{\theta}) + y (-\dot{\theta} / \cos \theta) = \mathbf{K}_1 \mathbf{v} + y \mathbf{K}_2 \mathbf{v}, \quad \forall y \in \mathbb{R} \quad (3.34)$$

FIG. 3.4 – Représentation (ρ, θ) des droites 2D

avec :

$$\begin{aligned} \mathbf{K}_1 &= (\lambda_1 \cos \theta \quad \lambda_1 \sin \theta \quad -\lambda_1 \rho \quad \sin \theta \quad -\cos \theta - \rho^2 / \cos \theta \quad -\rho \tan \theta) \\ \mathbf{K}_2 &= (\lambda_2 \cos \theta \quad \lambda_2 \sin \theta \quad -\lambda_2 \rho \quad \rho \quad \rho \tan \theta \quad 1 / \cos \theta) \end{aligned}$$

où $\lambda_1 = -\alpha \rho / \cos \theta - \gamma$ et $\lambda_2 = \alpha \tan \theta - \beta$.

On en déduit aussitôt :

$$\begin{cases} \dot{\rho} = (\mathbf{K}_1 + \rho \sin \theta \mathbf{K}_2) \mathbf{v} \\ \dot{\theta} = -\cos \theta \mathbf{K}_2 \mathbf{v} \end{cases} \quad (3.35)$$

d'où :

$$\begin{aligned} \mathbf{L}_\rho &= (\lambda_\rho \cos \theta \quad \lambda_\rho \sin \theta \quad -\lambda_\rho \rho \quad (1 + \rho^2) \sin \theta \quad -(1 + \rho^2) \cos \theta \quad 0) \\ \mathbf{L}_\theta &= (\lambda_\theta \cos \theta \quad \lambda_\theta \sin \theta \quad -\lambda_\theta \rho \quad -\rho \cos \theta \quad -\rho \sin \theta \quad -1) \end{aligned} \quad (3.36)$$

avec $\lambda_\rho = -\alpha \rho \cos \theta - \beta \rho \sin \theta - \gamma$ et $\lambda_\theta = -\alpha \sin \theta + \beta \cos \theta$. On peut retrouver le même résultat en appliquant l'équation (3.33) à deux points de \mathcal{D} , par exemple ceux de coordonnées $(\rho \cos \theta, \rho \sin \theta)$ et $(\rho \cos \theta + \sin \theta, \rho \sin \theta - \cos \theta)$. La matrice d'interaction associée à la mesure θ présente de bonnes propriétés de découplage autour d'une position désirée de la caméra telle que son plan image soit parallèle au plan objet. En effet, la matrice d'interaction associée à θ pour cette position peut s'écrire :

$$\mathbf{L}_\theta = (0 \quad 0 \quad 0 \quad -\rho \cos \theta \quad -\rho \sin \theta \quad -1) \quad (3.37)$$

On peut noter également le lien direct entre θ et la rotation autour de l'axe optique. Les résultats pour des primitives plus complexes (cercles, sphères et cylindres) sont donnés en [Chaumette 90], ce qui permet donc d'utiliser en asservissement visuel des informations 2D associées à ces primitives. Il est de plus possible d'en déduire la matrice d'interaction associée à des informations définies à partir de plusieurs primitives (par exemple l'orientation d'un segment ou la distance d'un point à une droite).

3.3.3 Matrice d'interaction 3D

Comme cela a déjà été évoqué, il est également possible de choisir des informations visuelles exprimées non plus directement dans l'image, mais issues d'une phase de reconstruction ou de localisation 3D [Wilson 96, Martinet 97]. L'obtention de ces informations 3D s'effectue soit par simple triangulation si l'on dispose d'un système de vision stéréoscopique calibré, soit, dans le cas d'un capteur monoculaire, par vision dynamique ou par une méthode de calcul de pose [Horaud 95]. Les techniques de vision dynamique reposent sur la mesure du mouvement de la caméra et du mouvement induit dans l'image. Elles sont en général assez sensibles aux erreurs de mesure [Xie 89, Boukir 93, Smith 94, Chaumette 96]. Nous donnons à présent la matrice d'interaction associée à la représentation minimale $\theta \mathbf{u}$ d'une rotation quelconque d'angle θ autour d'un axe \mathbf{u} , puis celle associée aux coordonnées d'un point 3D.

3.3.3.1 Matrice d'interaction associée à $\theta \mathbf{u}$

On rappelle tout d'abord que la représentation $\theta \mathbf{u}$ s'obtient de manière unique à partir des coefficients $r_{ij}(i=1..3, j=1..3)$ d'une matrice de rotation \mathbf{R} à l'aide de l'équation suivante [Khalil 99, Malis 98] :

$$\theta \mathbf{u} = \frac{1}{2 \operatorname{sinc} \theta} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix} \quad (3.38)$$

où $\theta = \arccos((r_{11} + r_{22} + r_{33} - 1)/2)$ et où le sinus cardinal $\operatorname{sinc} \theta$, défini par $\sin \theta = \theta \operatorname{sinc} \theta$, est une fonction C^∞ s'annulant en $(2n + 1)\pi, \forall n \in \mathbb{Z}$. Pour $\theta = \pi$, seul cas non pris en compte par (3.38), \mathbf{u} est le vecteur propre de \mathbf{R} associé à la valeur propre 1.

La matrice d'interaction associée à $\theta \mathbf{u}$ est donnée par [Malis 98] :

$$\mathbf{L}_{\theta \mathbf{u}} = \begin{pmatrix} \mathbf{0}_3 & \mathbf{L}_\omega \end{pmatrix} \quad (3.39)$$

avec :

$$\mathbf{L}_\omega = \mathbb{I}_3 - \frac{\theta}{2} [\mathbf{u}]_\times + \left(1 - \frac{\operatorname{sinc} \theta}{\operatorname{sinc}^2 \frac{\theta}{2}}\right) [\mathbf{u}]_\times^2 \quad (3.40)$$

La représentation $\theta \mathbf{u}$ est donc particulièrement intéressante puisque \mathbf{L}_ω est singulière uniquement pour $\theta = 2\pi$. Par ailleurs, on a :

$$\mathbf{L}_\omega^{-1} = \mathbb{I}_3 + \frac{\theta}{2} \operatorname{sinc}^2 \frac{\theta}{2} [\mathbf{u}]_\times + (1 - \operatorname{sinc} \theta) [\mathbf{u}]_\times^2 \quad (3.41)$$

et

$$\mathbf{L}_\omega^{-1} \theta \mathbf{u} = \theta \mathbf{u} \quad (3.42)$$

L'utilisation de $\theta \mathbf{u}$ permet également d'avoir des mesures invariantes aux mouvements de translation pour commander les 3 degrés de liberté de rotation.

3.3.3.2 Matrice d'interaction associée à un point 3D

A partir de l'équation fondamentale de la cinématique rappelée en (3.23), on obtient immédiatement pour tout point de coordonnées \mathbf{X} lié à l'objet :

$$\mathbf{L}_{\mathbf{X}} = \left(-\mathbb{I}_3 \quad [\mathbf{X}]_{\times} \right) \quad (3.43)$$

Les points pris en compte peuvent être des points caractéristiques de l'objet [Martinet 96], ou encore l'origine de R_o (on a alors $\mathbf{X} = {}^c\mathbf{P}_o$). À partir de la matrice d'interaction associée à \mathbf{X} , on peut noter que la variation de \mathbf{X} est linéairement dépendante des mouvements de translations. Dans le cas où le vecteur est exprimé dans un repère fixe, la matrice d'interaction associée à ce vecteur est donnée par :

$$\mathbf{L}_{\mathbf{X}} = \left(-\mathbb{I}_3 \quad 0 \right) \quad (3.44)$$

L'utilisation d'informations visuelles 3D permet donc d'avoir de bonnes propriétés de découplage, mais également d'avoir des matrices d'interaction moins non linéaires que dans le cas des mesures 2D. Dans l'hypothèse où les informations 3D sont estimées, cela permet de prouver facilement la stabilité du système. Cependant, si la caméra n'est pas parfaitement calibrée, ou si il y a des erreurs sur le modèle 3D de l'objet, les positions courante et désirée de la caméra ne sont pas correctement estimées. Si la pose désirée est calculée à partir d'une image acquise à cette position, ces erreurs d'estimations sont compensées par une commande en boucle fermée en supposant que le système converge bien entendu. Toutefois, les erreurs de calibration influencent énormément le comportement du système, et cela peut poser des problèmes de convergence si le déplacement considéré est très large. Le problème de stabilité au bruit de mesure se pose également, ainsi que les problèmes de visibilité de l'objet dans le cas où le déplacement à réaliser est composé de grands mouvements de translation et de rotation.

3.3.3.3 Informations visuelles 2D 1/2

La prise en compte d'informations visuelles 3D repose sur l'hypothèse que ces données peuvent être mesurées de manière fiable. En pratique, elles sont plus sensibles aux erreurs de mesure que les informations visuelles 2D, puisqu'obtenues à partir de celles-ci et d'un calcul de pose n'ayant pas de propriété particulière de lissage. Il est donc intéressant de combiner des informations visuelles 2D et 3D pour gagner en robustesse vis-à-vis des erreurs de mesure tout en conservant de bonnes propriétés de découplage. Dans [Malis 98, Malis 99], la fonction de tâche est définie de la manière suivante :

$$\mathbf{e} = \left(x - x^*, \quad y - y^*, \quad \log(Z/Z^*), \quad \theta \mathbf{u} \right) \quad (3.45)$$

où :

- (x, y) et (x^*, y^*) sont respectivement les coordonnées courante et désirée d'un point caractéristique dans l'image ;
- Z/Z^* est le rapport entre les distances courante et désirée de ce point ;
- $\theta \mathbf{u}$ représente la rotation à réaliser.

Dans ce cas, la matrice d'interaction est donnée par :

$$\mathbf{L}_s = \begin{pmatrix} \frac{1}{Z} \mathbf{L}_{e_v} & \mathbf{L}_{e_{v\omega}} \\ \mathbf{0}_3 & \mathbf{L}_\omega \end{pmatrix} \quad (3.46)$$

où \mathbf{L}_ω est donnée par (3.40) et (voir (3.25) et (3.43)) :

$$\mathbf{L}_{e_v} = \begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \\ 0 & 0 & -1 \end{pmatrix} \quad \text{et} \quad \mathbf{L}_{e_{v\omega}} = \begin{pmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \\ -y & x & 0 \end{pmatrix}$$

On peut ensuite choisir dans la commande :

$$\widehat{\mathbf{L}}_s^{-1} = \begin{pmatrix} Z\mathbf{L}_{e_v}^{-1} & -Z\mathbf{L}_{e_v}^{-1}\mathbf{L}_{e_{v\omega}} \\ \mathbf{0}_3 & \mathbb{I}_3 \end{pmatrix} \mathbf{e} \quad (3.47)$$

car on rappelle que $\mathbf{L}_\omega^{-1} \theta \mathbf{u} = \theta \mathbf{u}$ (voir ((3.42)) Un découplage satisfaisant est donc obtenu puisque la matrice de commande est triangulaire. De plus, la partie de cette matrice qui correspond aux mouvements de rotations vaut l'identité d'après (3.42). Cela permet en pratique d'obtenir des bonnes trajectoires 3D de la caméra et d'étudier facilement la stabilité du système. Cependant, comme cela été mentionné au début du chapitre 1 de ce manuscrit, ce type d'asservissement visuel est relativement sensible aux bruits de mesure.

3.4 Matrice d'interaction associée au moment

Dans la suite, nous allons déterminer la matrice d'interaction associée aux moments calculés à partir d'une distribution continue et à partir d'un ensemble discret de points. Nous verrons que les résultats obtenus sont similaires, mais présentent tout de même de légères différences. Nous traitons également deux sortes de projection dans l'image : tout d'abord la projection perspective classique sur un plan, puis la projection perspective sur une sphère de rayon égale à la distance focale de la caméra.

3.4.1 Modélisation de l'objet

On suppose dans la suite que l'objet est constitué d'une surface continue. Dans ce cas, la profondeur Z de chaque point 3D peut être exprimée en fonction des coordonnées x et y de sa projection dans l'image. Plus précisément, on pose :

$$\frac{1}{Z} = \sum_{p \geq 0, q \geq 0} A_{pq} x^p y^q \quad (3.48)$$

Dans le cas où l'objet est plan ou a une surface des limbes plane (voir figure 3.3), son équation dans le repère caméra peut être exprimée par :

$$Z = \gamma_1 X + \gamma_2 Y + Z_0,$$

En utilisant les équations de la projection perspective :

$$x = \frac{X}{Z}, \quad y = \frac{Y}{Z}$$

on en déduit :

$$\frac{1}{Z} = Ax + By + C \quad (3.49)$$

avec $A = -\gamma_1/Z_0$, $B = -\gamma_2/Z_0$ et $C = 1/Z_0$. Dans la suite de ce chapitre, nous allons nous limiter au cas où l'objet est plan. Les résultats obtenus peuvent être facilement généralisés au cas où la surface définissant l'objet est donnée par l'équation (3.48) pour une projection perspective [Chaumette 04].

3.4.2 Matrice d'interaction des moments calculés à partir d'une projection perspective

3.4.2.1 Matrice d'interaction des moments 2D : cas continu

Dans ce paragraphe, nous rappelons la matrice d'interaction des moments 2D donnée dans [Chaumette 02b]. Dans ce qui suit nous supposons que l'image est binaire, ou que le niveau de gris ne change pas si la caméra bouge (c'est-à-dire $\frac{df(x,y)}{dt} = 0$). Les moments 2D sont définis par :

$$m_{ij}(t) = \iint_{\mathcal{R}(t)} h(x, y) dx dy \quad (3.50)$$

où $h(x, y) = x^i y^j f(x, y)$. La seule partie de cette intégrale qui dépend du temps est le domaine d'intégration $\mathcal{R}(t)$. La variation temporelle de m_{ij} peut être calculée à partir de la variation du contour $\mathcal{C}(t)$. Plus précisément, on a [Hayt 89]:

$$\dot{m}_{ij} = \oint_{\mathcal{C}(t)} h(x, y) \dot{\mathbf{x}}^T \mathbf{n} dl \quad (3.51)$$

où $\dot{\mathbf{x}}$ est la vitesse d'un point $\mathbf{x} = (x, y)$ du contour, \mathbf{n} est le vecteur unitaire normal à $\mathcal{C}(t)$ en un point \mathbf{x} et dl un élément infinitésimal du contour $\mathcal{C}(t)$. D'un point de vue géométrique (voir figure 3.5), la variation de m_{ij} est donnée par la valeur du moment m_{ij} calculé sur la surface infinitésimale comprise entre $\mathcal{C}(t)$ et $\mathcal{C}(t + dt)$. Cette valeur n'est rien d'autre que l'intégrale le long de $\mathcal{C}(t)$ du produit de h par le produit scalaire entre $\dot{\mathbf{x}}$ et \mathbf{n} .

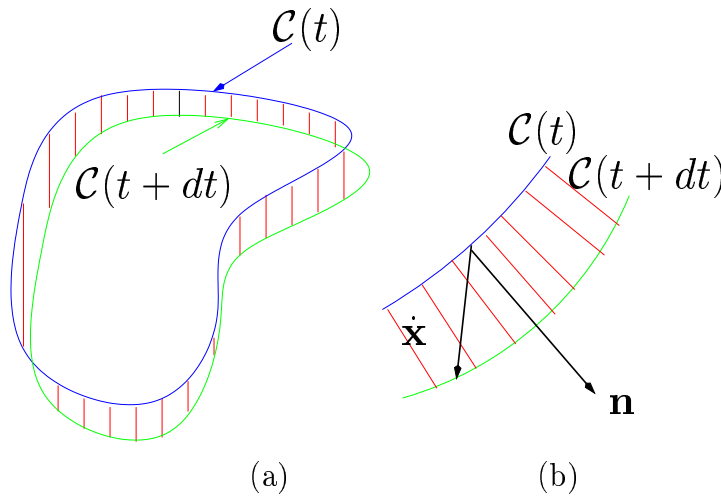


FIG. 3.5 – a) variation temporelle du contour $\mathcal{C}(t)$, b) détails

Si les conditions suivantes sont satisfaites (ce qui est le cas en pratique) :

1. $\mathcal{C}(t)$ est continue par partie

2. le vecteur $h(x, y) \dot{\mathbf{x}}$ est tangent à $\mathcal{R}(t)$ et différentiable continuellement, $\forall \mathbf{x} \in \mathcal{R}(t)$,

on peut utiliser le théorème de Green [Stewart 91] et écrire (3.51) sous la forme

$$\dot{m}_{ij} = \iint_{\mathcal{R}(t)} \operatorname{div}(h(x, y) \dot{\mathbf{x}}) dx dy \quad (3.52)$$

Après développement de (3.52), on obtient :

$$\dot{m}_{ij} = \iint_{\mathcal{R}} \left(\frac{\partial h}{\partial x} \dot{x} + \frac{\partial h}{\partial y} \dot{y} + h(x, y) \left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) \right) dx dy \quad (3.53)$$

On retrouve ainsi le résultat obtenu dans le chapitre précédent en utilisant une méthode basée sur un calcul variationnel (voir (2.46)). En utilisant (3.49) dans (3.25) on obtient :

$$\begin{cases} \dot{x} = -(Ax + By + C).v_x \\ \quad + x(Ax + By + C).v_z \\ \quad + xy.\omega_x - (1 + x^2).\omega_y + y.\omega_z \\ \dot{y} = -(Ax + By + C).v_y \\ \quad + y(Ax + By + C).v_z \\ \quad + (1 + y^2).\omega_x - xy.\omega_y - x.\omega_z \end{cases} \quad (3.54)$$

À partir de cette équation, on en déduit :

$$\begin{cases} \frac{\partial \dot{x}}{\partial x} = -Av_x + (2Ax + By + C)v_z + y\omega_x - 2x\omega_y \\ \frac{\partial \dot{y}}{\partial y} = -Bv_y + (Ax + 2By + C)v_z + 2y\omega_x - x\omega_y \end{cases} \quad (3.55)$$

Enfin, on a $\frac{\partial h}{\partial x} = ix^{i-1}y^j f(x, y)$ et $\frac{\partial h}{\partial y} = jx^i y^{j-1} f(x, y)$. La matrice d'interaction associée au moment m_{ij} peut alors être déterminée, et on obtient :

$$\mathbf{L}_{m_{ij}} = (m_{vx} \ m_{vy} \ m_{vz} \ m_{wx} \ m_{wy} \ m_{wz}) \quad (3.56)$$

où :

$$\begin{cases} m_{vx} = -i(Am_{ij} + Bm_{i-1,j+1} + Cm_{i-1,j}) - Am_{ij} \\ m_{vy} = -j(Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) - Bm_{ij} \\ m_{vz} = (i+j+3)(Am_{i+1,j} + Bm_{i,j+1} + Cm_{ij}) - Cm_{ij} \\ m_{wx} = (i+j+3)m_{i,j+1} + jm_{i,j-1} \\ m_{wy} = -(i+j+3)m_{i+1,j} - im_{i-1,j} \\ m_{wz} = im_{i-1,j+1} - jm_{i+1,j-1} \end{cases}$$

Matrice d'interaction des moments 2D d'ordre 0 et 1

À partir de la forme générale donnée par (3.56), on peut déduire la matrice d'interaction associée à un moment en sélectionnant la valeur de i et j . On considère par exemple le cas le plus simple : $i = j = 0$, qui donne la matrice d'interaction de la surface de l'objet $a = m_{00}$. On obtient directement de (3.56):

$$\mathbf{L}_a = \begin{pmatrix} -aA & -aB & a(3/Z_g - C) & 3ay_g & -3ax_g & 0 \end{pmatrix} \quad (3.57)$$

où $1/Z_g = Ax_g + By_g + C$. On peut noter que, si l'objet est centré et parallèle au plan image ($A = B = x_g = y_g = 0$), on a alors $\dot{a} = 0$ pour tous les degrés de liberté sauf la translation selon l'axe optique. La surface est donc particulièrement utile pour contrôler ce mouvement. D'une façon similaire, en considérant les moments d'ordre 1, m_{10} et m_{01} , on peut obtenir facilement les matrices d'interaction associées aux coordonnées du centre de gravité de l'objet dans l'image $x_g (= m_{10}/m_{00})$ et $y_g (= m_{01}/m_{00})$. On obtient :

$$\begin{aligned} \mathbf{L}_{x_g} &= \begin{pmatrix} -1/Z_g & 0 & x_{gvz} & x_{gwx} & x_{gwy} & y_g \end{pmatrix} \\ \mathbf{L}_{y_g} &= \begin{pmatrix} 0 & -1/Z_g & y_{gvz} & y_{gwx} & y_{gwy} & -x_g \end{pmatrix} \end{aligned} \quad (3.58)$$

où:

$$\begin{cases} x_{gvz} &= x_g/Z_g + 4(An_{20} + Bn_{11}) \\ y_{gvz} &= y_g/Z_g + 4(An_{11} + Bn_{02}) \\ x_{gwx} &= -y_{gwy} = x_g y_g + 4n_{11} \\ x_{gwy} &= -(1 + x_g^2 + 4n_{20}) \\ y_{gwx} &= 1 + y_g^2 + 4n_{02} \end{cases}$$

n_{20} , n_{02} et n_{11} sont les moments centrés normalisés d'ordre 2 :

$$n_{ij} = \mu_{ij}/a \text{ avec } \begin{cases} \mu_{20} = m_{20} - ax_g^2 \\ \mu_{02} = m_{02} - ay_g^2 \\ \mu_{11} = m_{11} - ax_g y_g \end{cases} \quad (3.59)$$

Matrice d'interaction des moments 2D centrés

De la même façon que pour les moments classiques, la matrice d'interaction associée au moment centré μ_{ij} peut être calculée. On obtient :

$$\mathbf{L}_{\mu_{ij}} = \begin{pmatrix} \mu_{vx} & \mu_{vy} & \mu_{vz} & \mu_{wx} & \mu_{wy} & \mu_{wz} \end{pmatrix} \quad (3.60)$$

avec :

$$\begin{aligned}
\mu_{vx} &= -(i+1)A\mu_{ij} - iB\mu_{i-1,j+1} \\
\mu_{vy} &= -jA\mu_{i+1,j-1} - (j+1)B\mu_{ij} \\
\mu_{vz} &= -A\mu_{wy} + B\mu_{wx} + (i+j+2)C\mu_{ij} \\
\mu_{wx} &= (i+j+3)\mu_{i,j+1} + ix_g\mu_{i-1,j+1} \\
&\quad + (i+2j+3)y_g\mu_{ij} - i4n_{11}\mu_{i-1,j} - j4n_{02}\mu_{i,j-1} \\
\mu_{wy} &= -(i+j+3)\mu_{i+1,j} - (2i+j+3)x_g\mu_{ij} \\
&\quad - jy_g\mu_{i+1,j-1} + i4n_{20}\mu_{i-1,j} + j4n_{11}\mu_{i,j-1} \\
\mu_{wz} &= i\mu_{i-1,j+1} - j\mu_{i+1,j-1}
\end{aligned}$$

où $n_{ij} = \mu_{ij}/m_{00}$. Ce résultat s'obtient par application directe du théorème de Green sur la définition des moments centrés donnés par (2.4). En effet, on obtient :

$$\begin{aligned}
\mu_{ij} &= \iint_{\mathcal{R}} \left(i(x-x_g)^{i-1}(y-y_g)^j(\dot{x}-\dot{x}_g) \right. \\
&\quad \left. + j(x-x_g)^i(y-y_g)^{j-1}(\dot{y}-\dot{y}_g) \right. \\
&\quad \left. + (x-x_g)^i(y-y_g)^j \left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) \right) dx dy
\end{aligned} \tag{3.61}$$

À partir de ce résultat et en utilisant (3.25) et (3.58), on retrouve (3.60). Pour les positions où l'objet est parallèle au plan image (c'est-à-dire $A = B = 0$), on note facilement à partir des deux premiers éléments de $\mathbf{L}_{\mu_{ij}}$ que les variations des moments centrés par rapport aux vitesses v_x et v_y sont nuls. Cela prouve leur invariance par rapport aux translations 2D parallèles au plan image lorsque l'objet est parallèle au plan image. On peut montrer également que si le plan image et le plan objet sont parallèles, les moments invariants aux changements d'échelle sont invariants par rapport aux translations suivant v_z . À titre d'exemple, si on considère :

$$r = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{20}^2 + 2\mu_{20}\mu_{02} + \mu_{02}^2} \tag{3.62}$$

alors on peut montrer en utilisant (3.60) que les trois premiers termes de la matrice d'interaction L_{rv_x} , L_{rv_y} et L_{rv_z} sont nuls quand $A = B = 0$. Même si ces invariances sont locales (c'est-à-dire valables seulement si $A = B = 0$), ces primitives varient principalement pour des rotations autour de la position désirée. Elles seront donc utilisées pour commander les rotations.

Appliquons à présent le résultat (3.60) sur les moments d'ordre 2, qui sont μ_{20} , μ_{02} et μ_{11} (on rappelle que les moments d'ordre 1 μ_{10} et μ_{01} sont nuls), on obtient alors :

$$\begin{aligned}
\mathbf{L}_{\mu_{20}} &= \begin{pmatrix} \mu_{20vx} & -B\mu_{20} & u_{20} & t_{20} & s_{20} & 2\mu_{11} \end{pmatrix} \\
\mathbf{L}_{\mu_{02}} &= \begin{pmatrix} -A\mu_{02} & \mu_{02vy} & u_{02} & t_{02} & s_{02} & -2\mu_{11} \end{pmatrix} \\
\mathbf{L}_{\mu_{11}} &= \begin{pmatrix} \mu_{11vx} & \mu_{11vy} & u_{11} & t_{11} & s_{11} & \mu_{02} - \mu_{20} \end{pmatrix}
\end{aligned} \tag{3.63}$$

où:

$$\left\{ \begin{array}{l} \mu_{20vx} = -3A\mu_{20} - 2B\mu_{11} \\ \mu_{02vy} = -2A\mu_{11} - 3B\mu_{02} \\ \mu_{11vx} = -2A\mu_{11} - B\mu_{02} \\ \mu_{11vy} = -2B\mu_{11} - A\mu_{20} \\ u_{ij} = -As_{ij} + Bt_{ij} + 4C\mu_{ij} \\ s_{20} = -7x_g\mu_{20} - 5\mu_{30} \\ t_{20} = 5(y_g\mu_{20} + \mu_{21}) + 2x_g\mu_{11} \\ s_{02} = -5(x_g\mu_{02} + \mu_{12}) - 2y_g\mu_{11} \\ t_{02} = 7y_g\mu_{02} + 5\mu_{03} \\ s_{11} = -6x_g\mu_{11} - 5\mu_{21} - y_g\mu_{20} \\ t_{11} = 6y_g\mu_{11} + 5\mu_{12} + x_g\mu_{02} \end{array} \right.$$

À partir des moments d'ordre 2, on peut déterminer une primitive intéressante, à savoir l'orientation de l'objet α définie par (voir [Mukundan 98] et Figure 3.6):

$$\alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \quad (3.64)$$

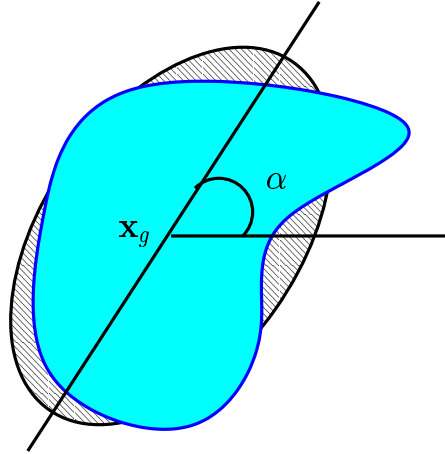


FIG. 3.6 – Orientation d'un objet et définition des paramètres usuels d'une ellipse

On a $\frac{d\arctan(x)}{dt} = \frac{\dot{x}}{1+x^2}$, d'où on peut déduire:

$$\mathbf{L}_\alpha = ((\mu_{20} - \mu_{02})\mathbf{L}_{\mu_{11}} - \mu_{11}(\mathbf{L}_{\mu_{20}} - \mathbf{L}_{\mu_{02}}))/\Delta$$

où $\Delta = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$. En utilisant (3.63), on obtient :

$$\mathbf{L}_\alpha = \begin{pmatrix} \alpha_{vx} & \alpha_{vy} & \alpha_{vz} & \alpha_{wx} & \alpha_{wy} & -1 \end{pmatrix} \quad (3.65)$$

$$\text{avec : } \begin{cases} \alpha_{vx} &= aA + bB \\ \alpha_{vy} &= -cA - aB \\ \alpha_{vz} &= -A\alpha_{wy} + B\alpha_{wx} \\ \alpha_{wx} &= -bx_g + ay_g + d \\ \alpha_{wy} &= ax_g - cy_g + e \end{cases}$$

et :

$$\begin{cases} a &= \mu_{11}(\mu_{20} + \mu_{02})/\Delta \\ b &= (2\mu_{11}^2 + \mu_{02}(\mu_{02} - \mu_{20}))/\Delta \\ c &= (2\mu_{11}^2 + \mu_{20}(\mu_{20} - \mu_{02}))/\Delta \\ d &= 5(\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21}))/\Delta \\ e &= 5(\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12}))/\Delta \end{cases}$$

On peut noter que α est invariante vis-à-vis des mouvements de translation selon les trois axes de la caméra pour une position de la caméra telle que son plan image soit parallèle à l'objet ($\alpha_{vx} = \alpha_{vy} = \alpha_{vz} = 0$ si $A = B = 0$). Par ailleurs, on peut noter également le lien direct entre la variation de α et le mouvement de rotation autour de l'axe optique de la caméra ω_z .

Nous verrons dans la section 3.5 de ce chapitre comment déterminer six combinaisons de moments pour contrôler les 6 degrés de liberté du système. Avec les six moments d'ordre inférieur ou égal à 2 ($a, x_g, y_g, \mu_{20}, \mu_{11}, \mu_{02}$) on peut montrer facilement que la matrice d'interaction associée est au plus de rang 5 si la forme de l'objet dans l'image est telle que ses moments centrés d'ordre 3 sont tous nuls (ce qui est le cas pour un rectangle par exemple). En considérant des objets de forme plus complexe, la matrice d'interaction est alors de rang 6, mais son conditionnement est médiocre et ne permet pas d'obtenir des résultats satisfaisants.

Nous déterminons à présent la forme analytique de la matrice d'interaction si l'on considère un ensemble de points pour calculer les moments [Tahri 04].

3.4.2.2 Matrice d'interaction des moments 2D : cas discret

Dans la suite de ce paragraphe, nous déterminons la matrice d'interaction associée aux moments calculés à partir d'un ensemble de points, puis nous comparons ce résultat avec celui obtenu dans le cas continu.

En dérivant l'équation (2.73) qui définit les moments 2D à partir de la position d'un ensemble de N points, on obtient :

$$\dot{m}_{ij} = \sum_{h=0}^N (ix_k^{i-1}y_k^j \dot{x}_k + jx_k^i y_k^{j-1} \dot{y}_k) \quad (3.66)$$

En comparant cette équation avec (3.53), on peut noter que (3.66) ne fait pas apparaître les termes $\frac{\partial \dot{x}}{\partial x}$ et $\frac{\partial \dot{y}}{\partial y}$. La matrice d'interaction sera donc différente dans les cas continu et discret. Si l'objet est plan, alors en combinant l'équation du plan

(3.49) et (3.25), on obtient :

$$\begin{cases} \dot{x}_k = -(Ax_k + By_k + C)v_x \\ \quad + x_k(Ax_k + By_k + C)v_z \\ \quad + x_k y_k \omega_x - (1 + x_k^2)\omega_y + y_k \omega_z \\ \dot{y}_k = -(Ax_k + By_k + C)v_y \\ \quad + y_k(Ax_k + By_k + C)v_z \\ \quad + (1 + y_k^2)\omega_x - x_k y_k \omega_y - x_k \omega_z \end{cases} \quad (3.67)$$

qui est strictement identique à (3.54). Finalement, à partir de (3.66) et (3.67), la matrice d'interaction associée au moment m_{ij} calculé à partir d'un ensemble de points peut être obtenu par de simples calculs. On obtient :

$$\mathbf{L}_{m_{ij}} = (m_{d_{vx}} \ m_{d_{vy}} \ m_{d_{vz}} \ m_{d_{wx}} \ m_{d_{wy}} \ m_{d_{wz}}) \quad (3.68)$$

où :

$$\begin{cases} m_{d_{vx}} = -i(Am_{ij} + Bm_{i-1,j+1} + Cm_{i-1,j}) \\ m_{d_{vy}} = -j(Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) \\ m_{d_{vz}} = (i+j)(Am_{i+1,j} + Bm_{i,j+1} + Cm_{ij}) \\ m_{d_{wx}} = (i+j)m_{i,j+1} + jm_{i,j-1} \\ m_{d_{wy}} = -(i+j)m_{i+1,j} - im_{i-1,j} \\ m_{d_{wz}} = im_{i-1,j+1} - jm_{i+1,j-1} \end{cases}$$

La matrice d'interaction obtenue n'est effectivement pas tout à fait la même que celle obtenue dans les cas d'une distribution continue (voir (3.56)). En effet, à partir des deux équations (3.53) et (3.66) donnant la variation des moments dans le cas continu et dans le cas discret, on peut noter que, même si les deux premiers termes des deux équations sont similaires, le troisième terme de (3.53) dû à la variation d'élément de surface autour d'un point de coordonnées (x, y) n'existe pas dans (3.66). Pour mieux illustrer cette différence, considérons le moment d'ordre 0. Dans le cas discret, le moment m_{00} est une constante (qui vaut N) quelle que soit la position de l'objet. On peut d'ailleurs vérifier en considérant $i = j = 0$ dans (3.68) que tous les termes de la matrice d'interaction $L_{m_{00}}$ sont égaux à 0, ce qui indique bien que m_{00} est invariant dans ce cas. Par contre, dans le cas continu, le moment m_{00} représente la surface de l'objet, cette dernière varie en fonction des mouvements du robot (sauf pour certains mouvements particuliers appartenant à $Ker(L_{m_{00}})$).

De la même manière que dans le cas continu, on dérive l'équation définissant les moments centrés (2.74), on obtient la formule de leur matrice d'interaction associée par :

$$\mathbf{L}_{\mu_{ij}} = (\mu_{d_{vx}} \ \mu_{d_{vy}} \ \mu_{d_{vz}} \ \mu_{d_{wx}} \ \mu_{d_{wy}} \ \mu_{d_{wz}}) \quad (3.69)$$

avec :

$$\left\{ \begin{array}{l} \mu_{d_{vx}} = -iA\mu_{ij} - iB\mu_{i-1,j+1} \\ \mu_{d_{vy}} = -jA\mu_{i+1,j-1} - jB\mu_{ij} \\ \mu_{d_{vz}} = -A\mu_{wy} + B\mu_{wx} + (i+j)C\mu_{ij} \\ \mu_{d_{wx}} = (i+j)\mu_{i,j+1} + ix_g\mu_{i-1,j+1} \\ \quad + (i+2j)y_g\mu_{ij} - in_{11}\mu_{i-1,j} - jn_{02}\mu_{i,j-1} \\ \mu_{d_{wy}} = -(i+j)\mu_{i+1,j} - (2i+j)x_g\mu_{ij} \\ \quad - jy_g\mu_{i+1,j-1} + in_{20}\mu_{i-1,j} + jn_{11}\mu_{i,j-1} \\ \mu_{d_{wz}} = i\mu_{i-1,j+1} - j\mu_{i+1,j-1} \end{array} \right.$$

où $n_{ij} = \mu_{ij}/m_{00}$. Comme dans le cas continu, si l'objet est parallèle au plan image (c'est-à-dire $A = B = 0$), on note à partir de $\mu_{d_{vx}}$ et $\mu_{d_{vy}}$ que les moments centrés sont invariants aux mouvements de translation parallèles au plan image ($\mu_{d_{vx}} = \mu_{d_{vy}} = 0$ si $A = B = 0$). On peut montrer également que les moments invariants aux changements d'échelle sont invariants au mouvement de translation selon l'axe optique.

3.4.3 Matrice d'interaction des moments calculés à partir d'une projection sphérique

La projection sphérique a été souvent utilisée en vision par ordinateur [Fermuller 00]. En asservissement visuel, son utilisation a été proposée tout récemment dans [Hamel 00], mais seulement pour les coordonnées d'un point. En fait, la projection sphérique présente de bonnes propriétés vis-à-vis des mouvements de rotation. En effet, un mouvement de rotation de la caméra se traduit par le même mouvement sur la projection sphérique de l'objet. Cette propriété permet à la projection de l'objet de garder la même forme quelque soit le mouvement de rotation. La détermination d'informations visuelles telles que les moments 3D invariants aux rotations s'avère donc intéressante pour réaliser un découplage des rotations et des translations. Ainsi, ces moments 3D calculés à partir d'une projection sphérique (tels que ceux nous avons obtenus dans le chapitre 2) seront également invariants aux mouvements de rotation. L'utilisation de telles informations visuelles permettra de réaliser un découplage entre mouvement de translation et mouvement de rotation.

Dans la suite, nous nous intéressons à la définition des moments définis à partir d'une projection sphérique. À partir de ces formules, nous calculons la matrice d'interaction associée à ces moments dans le cas d'une distribution continue et dans le cas discret.

3.4.3.1 Expression des moments en projection sphérique

On note par la suite x_s , y_s , et z_s les coordonnées de la projection d'un point de l'objet sur la sphère de rayon égal à 1 et dont le centre coïncide avec le centre optique de la caméra comme indiqué sur la figure 3.7.

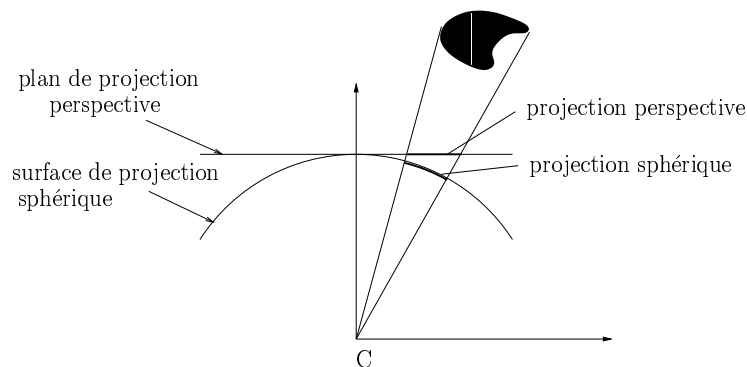


FIG. 3.7 – *Projection sphérique*

Projection d'un point sur la sphère unité

Le changement de variable faisant passer d'un point dans l'image plane à sa projection sur la sphère unité est donné par :

$$\begin{aligned} x_s &= \frac{x}{\sqrt{x^2+y^2+1}} = \frac{X}{\sqrt{X^2+Y^2+Z^2}} \\ y_s &= \frac{y}{\sqrt{x^2+y^2+1}} = \frac{Y}{\sqrt{X^2+Y^2+Z^2}} \\ z_s &= \frac{1}{\sqrt{x^2+y^2+1}} = \frac{Z}{\sqrt{X^2+Y^2+Z^2}} \end{aligned} \quad (3.70)$$

La dernière ligne correspond à la coordonnée z_s qui n'est plus constante et égale à 1 dans le cas sphérique. Réciproquement, les coordonnées (x, y) d'un point issu d'une projection perspective (x, y) peuvent être obtenues à partir des coordonnées d'un point en projection sphérique (x_s, y_s, z_s) par :

$$\begin{aligned} x &= \frac{x_s}{\sqrt{1 - (x_s^2 + y_s^2)}} \\ y &= \frac{y_s}{\sqrt{1 - (x_s^2 + y_s^2)}} \end{aligned} \quad (3.71)$$

Notons que la projection sphérique, bien que surfacique, n'est plus plane. Il semble donc intuitif d'utiliser des moments à trois dimensions, mais en utilisant comme élément d'intégration l'élément de surface de la projection sphérique. Les moments d'une projection sphérique peuvent alors être définis par :

$$m_{s_{i,j,k}} = \iint_{\mathcal{L}(t)} x_s^i y_s^j z_s^k f(x_s, y_s, z_s) ds \quad (3.72)$$

où $\mathcal{L}(t)$ est la surface de la sphère à un instant t .

Matrice d'interaction d'un point en projection sphérique

En dérivant l'équation (3.70), on obtient :

$$\begin{cases} \dot{x}_s = \frac{\dot{X}}{\sqrt{X^2+Y^2+Z^2}} - \frac{X}{\sqrt{X^2+Y^2+Z^2}} \frac{\dot{X}X+\dot{Y}Y+\dot{Z}Z}{X^2+Y^2+Z^2} \\ \dot{y}_s = \frac{\dot{Y}}{\sqrt{X^2+Y^2+Z^2}} - \frac{Y}{\sqrt{X^2+Y^2+Z^2}} \frac{\dot{X}X+\dot{Y}Y+\dot{Z}Z}{X^2+Y^2+Z^2} \\ \dot{z}_s = \frac{\dot{Z}}{\sqrt{X^2+Y^2+Z^2}} - \frac{Z}{\sqrt{X^2+Y^2+Z^2}} \frac{\dot{X}X+\dot{Y}Y+\dot{Z}Z}{X^2+Y^2+Z^2} \end{cases} \quad (3.73)$$

En remplaçant \dot{X} , \dot{Y} et \dot{Z} par leurs formules classiques (3.23), la matrice d'interaction d'un point en projection sphérique peut s'obtenir facilement. On obtient :

$$\mathbf{L}_{m_{s_{i,j,k}}} = \begin{pmatrix} -\frac{1}{r} + \frac{x_s^2}{r} & \frac{x_s y_s}{r} & \frac{x_s z_s}{r} & 0 & -z_s & y_s \\ \frac{x_s y_s}{r} & -\frac{1}{r} + \frac{y_s^2}{r} & \frac{y_s z_s}{r} & z_s & 0 & -x_s \\ \frac{x_s z_s}{r} & \frac{y_s z_s}{r} & -\frac{1}{r} + \frac{z_s^2}{r} & -y_s & x_s & 0 \end{pmatrix} \quad (3.74)$$

où $r = \sqrt{X^2 + Y^2 + Z^2}$. Si on considère un objet plan, la valeur de $\frac{1}{r}$ qui apparaît dans la matrice (3.74) peut s'écrire, en utilisant (3.49) et (3.70), en fonction des

coordonnées en projection sphérique et des paramètres du plan. Plus précisément, on a :

$$\begin{aligned} \frac{1}{r} &= \frac{1}{\sqrt{X^2 + Y^2 + Z^2}} = \frac{1}{Z} \frac{1}{\sqrt{x^2 + y^2 + 1}} \\ &= \frac{Ax}{\sqrt{x^2 + y^2 + 1}} + \frac{By}{\sqrt{x^2 + y^2 + 1}} + \frac{C}{\sqrt{x^2 + y^2 + 1}} \end{aligned} \quad (3.75)$$

On obtient alors :

$$\frac{1}{r} = Ax_s + By_s + Cz_s \quad (3.76)$$

Projection d'un élément de surface sur la sphère unité

Le but de ce paragraphe est d'exprimer la correspondance entre un pixel du plan image et l'élément de surface sphérique associé soit ds'' en fonction de ds (voir figure 3.8). Pour cela, nous utilisons l'intermédiaire ds' .

Pour des éléments de surface infinitésimaux, $ds' \vec{n}'$ est la projection de $ds \vec{n}$ sur la direction du point de vue. On obtient alors :

$$ds' = ds \cdot \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \cdot \frac{\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}}{\sqrt{x^2 + y^2 + 1}} = \frac{ds}{\sqrt{x^2 + y^2 + 1}} \quad (3.77)$$

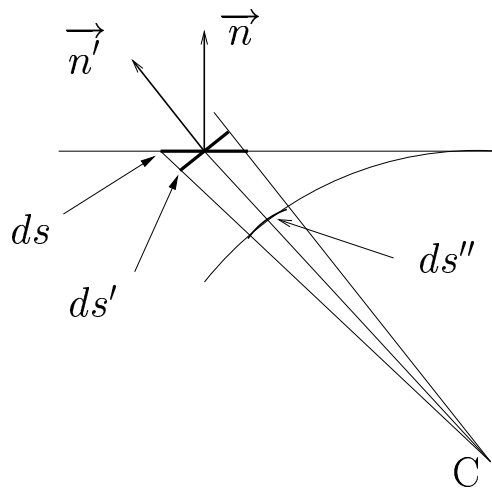


FIG. 3.8 – *Élément de surface en projection sphérique*

D'autre part, toujours sous l'hypothèse d'éléments infinitésimaux :

$$ds'' = \frac{ds'}{x^2 + y^2 + 1}$$

Le changement de variable concernant les éléments de surface est donc donné par :

$$ds'' = \frac{ds}{(x^2 + y^2 + 1)^{\frac{3}{2}}} \quad (3.78)$$

En combinant (3.78), (3.70) et (3.72), les moments d'une projection sphérique peuvent être calculés directement à partir de la projection sur le plan image par :

$$m_{s_{i,j,k}} = \iint_{\mathcal{R}} \left(\frac{x}{\sqrt{x^2 + y^2 + 1}} \right)^i \left(\frac{y}{\sqrt{x^2 + y^2 + 1}} \right)^j \left(\frac{1}{\sqrt{x^2 + y^2 + 1}} \right)^k f(x, y) \frac{dxdy}{(x^2 + y^2 + 1)^{\frac{3}{2}}} \quad (3.79)$$

ou encore :

$$m_{s_{i,j,k}} = \iint_{\mathcal{R}} x_s^i y_s^j z_s^{k+3} f(x, y) dxdy \quad (3.80)$$

Dans le cas où on traite un nuage de points, les moments sphériques sont donnés par :

$$m_{s_{i,j,k}} = \sum_{h=0}^N x_{s_h}^i y_{s_h}^j z_{s_h}^k \quad (3.81)$$

On peut noter la différence dans la définition des moments dans le cas continu et dans le cas discret : dans le cas continu, le terme z_s^3 apparaît pour prendre en compte l'élément de surface, ce qui n'intervient pas dans le cas discret. Comme nous allons le voir à présent, les matrices d'interaction seront donc différentes.

3.4.3.2 Matrice d'interaction des moments d'une projection sphérique : cas continu

On considère tout d'abord le cas continu. En dérivant l'équation (3.79) et en utilisant la même méthode que pour (3.53), on obtient :

$$\dot{m}_{s_{i,j,k}} = \iint_{\mathcal{R}} \left(\frac{\partial h}{\partial x} \dot{x} + \frac{\partial h}{\partial y} \dot{y} + h(x, y) \left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) \right) dxdy \quad (3.82)$$

où $h(x, y) = \left(\frac{x}{\sqrt{x^2 + y^2 + 1}} \right)^i \left(\frac{y}{\sqrt{x^2 + y^2 + 1}} \right)^j \left(\frac{1}{\sqrt{x^2 + y^2 + 1}} \right)^{k+3} f(x, y)$. Après développement de (3.82) et en combinant le résultat avec (3.53) et (3.54), on obtient :

$$\mathbf{L}_{m_{s_{i,j,k}}} = (m_{s_{vx}} \ m_{s_{vy}} \ m_{s_{vz}} \ m_{s_{wx}} \ m_{s_{wy}} \ m_{s_{wz}}) \quad (3.83)$$

où :

$$\begin{cases} m_{s_{vx}} = A(\beta m_{s_{i+2,j,k}} - (i+1)m_{s_{i,j,k}}) + B(\beta m_{s_{i+1,j+1,k}} - im_{s_{i-1,j+1,k}}) \\ \quad + C(\beta m_{s_{i+1,j,k+1}} - im_{s_{i-1,j,k+1}}) \\ m_{s_{vy}} = A(\beta m_{s_{i+1,j+1,k}} - jm_{s_{i+1,j-1,k}}) + B(\beta m_{s_{i,j+2,k}} - (j+1)m_{s_{i,j,k}}) \\ \quad + C(\beta m_{s_{i,j+1,k+1}} - jm_{s_{i,j-1,k+1}}) \\ m_{s_{vz}} = A(\beta m_{s_{i+1,j,k+1}} - km_{s_{i+1,j,k-1}}) + B(\beta m_{s_{i,j+1,k+1}} - km_{s_{i,j+1,k-1}}) \\ \quad + C(\beta m_{s_{i,j,k+2}} - (k+1)m_{s_{i,j,k}}) \\ m_{s_{wx}} = jm_{s_{i,j-1,k+1}} - km_{s_{i,j+1,k-1}} \\ m_{s_{wy}} = km_{s_{i+1,j,k-1}} - im_{s_{i-1,j,k+1}} \\ m_{s_{wz}} = im_{s_{i-1,j+1,k}} - jm_{s_{i+1,j-1,k}} \end{cases}$$

avec $\beta = i + j + k + 3$. À partir de cette matrice, on peut montrer immédiatement que la surface d'un objet sous projection sphérique est invariante au mouvement de rotation. En effet, la matrice d'interaction associée à $m_{s_{000}}$ est nulle pour les mouvements de rotation quand $i = j = k = 0$. D'autres moments invariants aux mouvements de rotation peuvent être obtenus à partir d'une projection sphérique en utilisant la méthode de calcul des invariants proposée au chapitre 2. Dans le paragraphe suivant, nous calculons la matrice d'interaction associée aux moments d'une projection sphérique d'un ensemble de points.

3.4.3.3 Matrice d'interaction des moments d'une projection sphérique : cas discret

On considère à présent le cas discret. En dérivant l'équation (3.81), on obtient :

$$m_{s_{i,j,k}} = \sum_{h=0}^N (i x_{s_h}^{i-1} y_{s_h}^j z_{s_h}^k \dot{x}_{s_h} + j x_{s_h}^i y_{s_h}^{j-1} z_{s_h}^k \dot{y}_{s_h} + k x_{s_h}^i y_{s_h}^j z_{s_h}^{k-1} \dot{z}_{s_h}) \quad (3.84)$$

En combinant cette équation avec (3.74) et (3.76), on obtient :

$$\mathbf{L}_{m_{s_{i,j,k}}} = (m_{s_{vx}} \ m_{s_{vy}} \ m_{s_{vz}} \ m_{s_{wx}} \ m_{s_{wy}} \ m_{s_{wz}}) \quad (3.85)$$

où :

$$\begin{cases} m_{s_{vx}} = A(\beta_d m_{s_{i+2,j,k}} - im_{s_{i,j,k}}) + B(\beta_d m_{s_{i+1,j+1,k}} - im_{s_{i-1,j+1,k}}) \\ \quad + C(\beta_d m_{s_{i+1,j,k+1}} - im_{s_{i-1,j,k+1}}) \\ m_{s_{vy}} = A(\beta_d m_{s_{i+1,j+1,k}} - jm_{s_{i+1,j-1,k}}) + B(\beta_d m_{s_{i,j+2,k}} - jm_{s_{i,j,k}}) \\ \quad + C(\beta_d m_{s_{i,j+1,k+1}} - jm_{s_{i,j-1,k+1}}) \\ m_{s_{vz}} = A(\beta_d m_{s_{i+1,j,k+1}} - km_{s_{i+1,j,k-1}}) + B(\beta_d m_{s_{i,j+1,k+1}} - km_{s_{i,j+1,k-1}}) \\ \quad + C(\beta_d m_{s_{i,j,k+2}} - km_{s_{i,j,k}}) \\ m_{s_{wx}} = jm_{s_{i,j-1,k+1}} - km_{s_{i,j+1,k-1}} \\ m_{s_{wy}} = km_{s_{i+1,j,k-1}} - im_{s_{i-1,j,k+1}} \\ m_{s_{wz}} = im_{s_{i-1,j+1,k}} - jm_{s_{i+1,j-1,k}} \end{cases}$$

avec $\beta_d = i + j + k$. On peut noter de nouveau que la matrice d'interaction associée aux moments sphériques dans le cas discret (3.85) est différente de celle calculée dans le cas continu (3.83). Cette différence s'explique exactement de la même façon que dans le cas des moments 2D (voir (3.82) et (3.84)).

Dans la section suivante, nous discutons du choix des informations visuelles en nous basant sur les différents résultats obtenus sur les moments invariants et sur les lois de commande par asservissement visuel.

3.5 Choix des informations visuelles

Dans cette section, nous déterminons six informations visuelles pour commander les six degrés de liberté du robot en nous basant sur les résultats théoriques présentés auparavant. Notre objectif est d'obtenir une matrice d'interaction creuse qui varie doucement autour de la position désirée. Dans le paragraphe suivant, nous étudierons l'utilisation des invariants aux rotations calculés à partir d'une projection sphérique pour contrôler les mouvements de translation du robot. Nous proposons par la suite des informations visuelles basées sur les invariants d'une projection plane pour obtenir une matrice triangulaire si l'objet est parallèle au plan image. En outre, pour ces positions, les éléments de la matrice d'interaction qui correspondent aux translations forment un bloc diagonal constant. Dans [Chaumette 02b], cette propriété n'était pas vérifiée. Dans la suite, nous supposons tout d'abord que la position désirée du plan image est parallèle à l'objet (c'est-à-dire $A = B = 0$) et on note \mathbf{L}_s^{\parallel} la matrice d'interaction qui correspond à cette position. Le cas où la position du plan image n'est pas parallèle à l'objet sera décrit ensuite au paragraphe 3.7. Pour l'instant, on discute la possibilité d'utiliser des invariants en projection sphérique pour obtenir des informations visuelles qui ne dépendent que des mouvements de translation.

3.5.1 Utilisation des invariants en projection sphérique

Les points issus d'une projection sphérique se comportent de la même manière que leur correspondant en 3D vis-à-vis des mouvements de rotation de la caméra. En effet, en utilisant (3.70), les coordonnées d'un point issu d'une projection sphérique s'écrivent après un mouvement de rotation :

$$\begin{aligned} x'_s &= \frac{X'}{\sqrt{X'^2 + Y'^2 + Z'^2}} \\ y'_s &= \frac{Y'}{\sqrt{X'^2 + Y'^2 + Z'^2}} \\ z'_s &= \frac{Z'}{\sqrt{X'^2 + Y'^2 + Z'^2}} \end{aligned} \tag{3.86}$$

avec :

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (3.87)$$

où les r_{ij} sont les éléments de la matrice de rotation. On en déduit :

$$\begin{aligned} x'_s &= \frac{r_{11}X + r_{12}Y + r_{13}Z}{\sqrt{X'^2 + Y'^2 + Z'^2}} \\ y'_s &= \frac{r_{21}X + r_{22}Y + r_{23}Z}{\sqrt{X'^2 + Y'^2 + Z'^2}} \\ z'_s &= \frac{r_{31}X + r_{32}Y + r_{33}Z}{\sqrt{X'^2 + Y'^2 + Z'^2}} \end{aligned} \quad (3.88)$$

Comme une transformation orthogonale ne change pas le module, on a :

$$X'^2 + Y'^2 + Z'^2 = X^2 + Y^2 + Z^2$$

on a alors :

$$\begin{aligned} x'_s &= \frac{r_{11}X + r_{12}Y + r_{13}Z}{\sqrt{X^2 + Y^2 + Z^2}} \\ y'_s &= \frac{r_{21}X + r_{22}Y + r_{23}Z}{\sqrt{X^2 + Y^2 + Z^2}} \\ z'_s &= \frac{r_{31}X + r_{32}Y + r_{33}Z}{\sqrt{X^2 + Y^2 + Z^2}} \end{aligned} \quad (3.89)$$

En combinant ce résultat avec (3.70), on obtient alors :

$$\begin{aligned} x'_s &= r_{11}x_s + r_{12}y_s + r_{13}z_s \\ y'_s &= r_{21}x_s + r_{22}y_s + r_{23}z_s \\ z'_s &= r_{31}x_s + r_{32}y_s + r_{33}z_s \end{aligned} \quad (3.90)$$

On retrouve donc le même effet des rotations pour un point issu d'une projection sphérique que pour un point 3D. Ce résultat peut être déduit également de l'équation (3.74). En effet, le bloc de la matrice d'interaction associée à un point en projection sphérique qui relie la variation des coordonnées de ce point aux vitesses de rotation est identique à celui associé à un point 3D. Les moments 3D invariants aux mouvements de rotation peuvent donc être utilisés pour obtenir des invariants aux rotations pour une projection sphérique. En utilisant alors la méthode proposée dans le chapitre précédent de ce manuscrit, plusieurs invariants aux mouvements de rotation peuvent être déterminés :

$$q(\mathbf{m}_0^1) = m_{000} \quad (3.91)$$

$$q(\mathbf{m}_2^1) = m_{200} + m_{020} + m_{002} = m_{000} \quad (3.92)$$

$$q_1(\mathbf{m}_2^2) = -m_{200}m_{020} - m_{200}m_{002} + m_{110}^2 + m_{101}^2 - m_{020}m_{002} + m_{011}^2 \quad (3.93)$$

$$q_2(\mathbf{m}_2^2) = m_{200}^2 + 2m_{200}m_{020} + 2m_{200}m_{002} + m_{020}^2 + 2m_{020}m_{002} + m_{002}^2 \quad (3.94)$$

$$= m_{000}^2 \quad (3.95)$$

Pour démontrer que $m_{200} + m_{020} + m_{002} = m_{000}$, il suffit de noter que $x_s^2 + y_s^2 + z_s^2 = 1$. D'autres polynômes invariants aux mouvements de rotation sont donnés en annexe de ce manuscrit. Les éléments de la matrice d'interaction associés à ces invariants qui correspondent aux mouvements de rotation sont bien évidemment nuls. Les valeurs numériques des matrices d'interaction associées aux polynômes invariants (A.27), (A.28), (A.29), (3.91), (A.16), (A.17), (A.21), (A.22), et (A.23) et calculées pour l'image donnée sur la figure 3.9 sont données par :

$$\mathbf{L}_{inv} = \begin{pmatrix} 0.0313 & -0.0296 & 1 & 0 & 0 \\ 0.0428 & -0.0263 & 1 & 0 & 0 \\ 0.0632 & -0.0218 & 1 & 0 & 0 \\ 0.0631 & -0.0218 & 1 & 0 & 0 \\ 0.0424 & -0.0264 & 1 & 0 & 0 \\ 0.0428 & -0.0263 & 1 & 0 & 0 \\ 0.0631 & -0.0218 & 1 & 0 & 0 \\ 0.0428 & -0.0263 & 1 & 0 & 0 \\ 0.0312 & -0.0297 & 1 & 0 & 0 \end{pmatrix} \quad (3.96)$$

En fait, ces valeurs ont été obtenues après normalisation par l'élément qui correspond à la translation selon l'axe optique de la caméra (d'où le 1 sur cette composante). On note à partir des valeurs numériques obtenues que tous les invariants dépendent principalement du mouvement selon l'axe optique. En effet, les variations de ces invariants par rapport aux mouvements selon les axes X et Y de la caméra sont négligeables devant celle par rapport au mouvement selon l'axe Z . L'utilisation de ces invariants pour contrôler les trois degrés de liberté de translation est donc impossible, étant donné le mauvais conditionnement de la matrice d'interaction obtenue. Nous n'avons donc pas retenu de tels invariants. Il serait possible d'en retenir un seul (par exemple m_{000}) pour contrôler la translation selon l'axe optique, mais nous avons préféré sélectionner six combinaisons de moments issus d'une projection perspective plane classique. Nous décrivons à présent cette sélection.



FIG. 3.9 – Image d'une "baleine"

3.5.2 Informations visuelles pour commander les translations

Dans [Corke 01,Chaumette 02b], les trois informations visuelles proposées pour commander les trois degrés de liberté de translation sont les coordonnées du centre de gravité (x_g, y_g) et la surface de l'objet a dans l'image. Dans ce cas quand les plans image et objet sont parallèles ($A = B = 0$), on obtient à partir de (3.57) et (3.58) :

$$\begin{aligned} \mathbf{L}_{x_g}^{\parallel} &= (-C \ 0 \ Cx_g \ \epsilon_1 \ -(1+\epsilon_2) \ y_g \) \\ \mathbf{L}_{y_g}^{\parallel} &= (0 \ -C \ Cy_g \ 1+\epsilon_3 \ -\epsilon_1 \ -x_g \) \\ \mathbf{L}_a^{\parallel} &= (0 \ 0 \ 2aC \ 3ay_g \ -3ax_g \ 0 \) \end{aligned} \quad (3.97)$$

avec $\epsilon_1 = n_{11} + x_g y_g$, $\epsilon_2 = n_{20} + x_g^2$ et $\epsilon_3 = n_{02} + y_g^2$. Même si cette matrice est triangulaire, on note la forte non linéarité de ses éléments. On peut aussi noter que les dynamiques ne sont pas les mêmes sur chaque axe de translation.

Notre choix est également basé sur ces informations visuelles intuitives mais en rajoutant une normalisation adéquate. Plus précisément, on définit :

$$a_n = Z^* \sqrt{\frac{a^*}{a}}, \quad x_n = a_n x_g, \quad y_n = a_n y_g \quad (3.98)$$

où a^* est la surface désirée de l'objet dans l'image et où Z^* est la profondeur désirée entre la caméra et l'objet. La matrice d'interaction associée à ces informations visuelles normalisées peut être facilement déterminée à partir de (3.97). En notant que, quand l'objet est parallèle au plan image, on a $Z^* \sqrt{a^*} = Z \sqrt{a} = \sqrt{S}$ où S est la surface de l'objet 3D, on obtient :

$$\begin{aligned} \mathbf{L}_{x_n}^{\parallel} &= (-1 \ 0 \ 0 \ a_n \epsilon_{11} \ -a_n(1+\epsilon_{12}) \ y_n \) \\ \mathbf{L}_{y_n}^{\parallel} &= (0 \ -1 \ 0 \ a_n(1+\epsilon_{21}) \ -a_n \epsilon_{11} \ -x_n \) \\ \mathbf{L}_{a_n}^{\parallel} &= (0 \ 0 \ -1 \ -3y_n/2 \ 3x_n/2 \ 0 \) \end{aligned} \quad (3.99)$$

avec $\epsilon_{11} = n_{11} - x_g y_g / 2$, $\epsilon_{12} = n_{20} - x_g^2 / 2$, et $\epsilon_{21} = n_{02} - y_g^2 / 2$. Puisque a_n est inversement proportionnelle à \sqrt{a} , on retrouve le résultat récent présenté dans [Mahony 02] qui montre qu'une telle primitive est proportionnelle à la profondeur (on note le terme constant dans le troisième élément de $\mathbf{L}_{a_n}^{\parallel}$). La normalisation par $Z^* \sqrt{a^*}$ n'a été faite que pour que ce terme constant soit égal à -1 . En outre, le choix de x_n et y_n permet un découplage total des 3 degrés de liberté de translation. On obtient aussi la même dynamique sur les trois axes de translation (notons le bloc diagonal égal à $-\mathbf{I}_3$ dans (3.99)). Cette bonne propriété permettra d'obtenir une bonne trajectoire des translations de la caméra. Finalement, on peut noter à partir de $\mathbf{L}_{x_n}^{\parallel}$ et de $\mathbf{L}_{y_n}^{\parallel}$, le couplage classique entre v_x et ω_y et entre v_y et ω_x . Ce couplage naturel permet en fait que l'objet reste dans le champ de vision de la caméra.

3.5.3 Informations visuelles pour commander les rotations

Pour contrôler l'orientation autour de l'axe optique, on a naturellement choisi, comme dans [Corke 01,Chaumette 02b], l'angle d'orientation de l'axe principal d'inertie de l'objet avec l'axe des abscisses du repère lié à l'image, soit :

$$\alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$$

À partir de (3.65), on obtient :

$$\mathbf{L}_\alpha^\parallel = (0 \ 0 \ 0 \ \alpha_{wx} \ \alpha_{wy} \ -1) \quad (3.100)$$

où :

$$\begin{cases} \alpha_{wx} = -bx_g + ay_g + d \\ \alpha_{wy} = ax_g - cy_g + e \end{cases}$$

et :

$$\begin{cases} a = \mu_{11}(\mu_{20} + \mu_{02})/\Delta \\ b = [2\mu_{11}^2 + \mu_{02}(\mu_{02} - \mu_{20})]/\Delta \\ c = [2\mu_{11}^2 + \mu_{20}(\mu_{20} - \mu_{02})]/\Delta \\ d = 5[\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21})]/\Delta \\ e = 5[\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12})]/\Delta \\ \Delta = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \end{cases}$$

On rappelle que α dépend essentiellement de la rotation ω_z autour de l'axe optique (nous verrons dans le chapitre 5 que les valeurs numériques de α_{wx} et α_{wy} sont faibles). Pour contrôler les deux derniers mouvements de rotation, nous employons deux combinaisons de moments r_i et r_j choisies parmi celles données ci-dessous :

$$\begin{aligned} r_1 &= \frac{I_{n1}}{I_{n2}}, \quad r_2 = \frac{I_{n3}}{I_{n4}}, \quad r_3 = \frac{I_{n5}}{I_{n6}}, \quad r_4 = \frac{I_{n7}}{I_{n6}}, \quad r_5 = \frac{I_{n8}}{I_{n6}}, \\ r_6 &= \frac{I_{n9}}{I_{n6}}, \quad r_7 = \frac{I_{n11}}{I_{n10}}, \quad r_8 = \frac{I_{n12}}{I_{n10}}, \quad r_9 = \frac{I_{n13}}{I_{n15}}, \quad r_{10} = \frac{I_{n14}}{I_{n15}} \end{aligned} \quad (3.101)$$

où

$$I_{n1} = -\mu_{20}\mu_{02} + \mu_{11}^2 \quad (3.102)$$

$$I_{n2} = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad (3.103)$$

$$I_{n3} = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})(3\mu_{21} - \mu_{03})^2 \quad (3.104)$$

$$I_{n_4} = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \quad (3.105)$$

$$I_{n_5} = -\mu_{30}^2\mu_{03}^2 + 6\mu_{30}\mu_{21}^3 - 4\mu_{30}\mu_{12}^3 - 4\mu_{21}^3\mu_{03} + 3\mu_{21}^2\mu_{12}^2 \quad (3.106)$$

$$I_{n_6} = 3\mu_{30}^2\mu_{12}^2 + 2\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}^2\mu_{12} - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 2\mu_{30}\mu_{12}^3 + 3\mu_{21}^4 \\ + 2\mu_{21}^3\mu_{03} + 3\mu_{21}^2\mu_{03}^2 - 6\mu_{21}\mu_{12}^2\mu_{03} + 3\mu_{12}^4 - 6\mu_{21}\mu_{12}^2\mu_{03} + 3\mu_{12}^4 \quad (3.107)$$

$$I_{n_7} = -\mu_{30}^3\mu_{03} + 3\mu_{30}^2\mu_{21}\mu_{12} - 2\mu_{30}\mu_{21}^3 - 3\mu_{30}\mu_{21}^2\mu_{03} + 6\mu_{30}\mu_{21}\mu_{12}^2 \\ + 3\mu_{30}\mu_{12}^2\mu_{03} + \mu_{30}\mu_{03}^3 - 3\mu_{21}^3\mu_{12} - 6\mu_{21}^2\mu_{12}\mu_{03} + 3\mu_{21}\mu_{12}^3 \\ - 3\mu_{21}\mu_{12}\mu_{03}^2 + 2\mu_{12}^3\mu_{03} \quad (3.108)$$

$$I_{n_8} = -\mu_{30}^3\mu_{12} + \mu_{30}^2\mu_{21}^2 - \mu_{30}^2\mu_{21}\mu_{03} - 2\mu_{30}^2\mu_{12}^2 + 3\mu_{30}\mu_{21}^2\mu_{12} \\ - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 3\mu_{30}\mu_{12}^3 - \mu_{30}\mu_{12}\mu_{03}^2 + 3\mu_{21}^3\mu_{03} + 3\mu_{21}^3\mu_{03} \\ - 2\mu_{21}^2\mu_{03}^2 + 3\mu_{21}\mu_{12}^2\mu_{03} - \mu_{21}\mu_{03}^3 + \mu_{12}^2\mu_{03}^2 \quad (3.109)$$

$$I_{n_9} = \mu_{30}^4 + 6\mu_{30}^3\mu_{12} + 6\mu_{30}^2\mu_{21}\mu_{03} + 9\mu_{30}^2\mu_{12}^2 + 2\mu_{30}^2\mu_{03}^2 + 18\mu_{30}\mu_{21}\mu_{12}\mu_{03} \\ + 6\mu_{30}\mu_{12}\mu_{03}^2 + 9\mu_{21}^2\mu_{03}^2 + 6\mu_{21}\mu_{03}^3 + \mu_{03}^4 \quad (3.110)$$

$$I_{n_{10}} = \mu_{40}\mu_{04} - 4\mu_{31}\mu_{13} + 3\mu_{22}\mu_{22} \quad (3.111)$$

$$I_{n_{11}} = 3\mu_{40}\mu_{22} - 2\mu_{40}\mu_{04} + 3\mu_{31}\mu_{31} + 2\mu_{31}\mu_{13} - 3\mu_{22}\mu_{04} + 3\mu_{13}\mu_{13} \quad (3.112)$$

$$I_{n_{12}} = 3\mu_{40}\mu_{40} + 12\mu_{40}\mu_{22} + 2\mu_{40}\mu_{04} + 16\mu_{31}\mu_{13} + 12\mu_{22}\mu_{04} + 3\mu_{04}\mu_{04} \quad (3.113)$$

$$I_{n_{13}} = (\mu_{50} + 2\mu_{32} + \mu_{14})^2 + (\mu_{05} + 2\mu_{23} + \mu_{41})^2 \quad (3.114)$$

$$I_{n_{14}} = (\mu_{50} - 2\mu_{32} - 3\mu_{14})^2 + (\mu_{05} - 2\mu_{23} - 3\mu_{41})^2 \quad (3.115)$$

$$I_{n_{15}} = (\mu_{50} - 10\mu_{32} + 5\mu_{14})^2 + (\mu_{05} - 10\mu_{23} + 5\mu_{41})^2 \quad (3.116)$$

Ces combinaisons sont invariantes aux mouvements de translation parallèles au plan image, aux changements d'échelles (aux translations selon l'axe optique si l'objet est parallèle au plan image), ainsi qu'au mouvement de rotation autour l'axe optique. On rappelle que ces invariants ont été obtenus grâce aux développements présentés dans le chapitre précédent. On peut également obtenir des invariants de même type en utilisant une normalisation par la surface. Cependant, les valeurs des invariants pouvant être obtenus par cette normalisation ont des valeurs très faibles, ce qui pose des problèmes numériques. En effet, les valeurs des polynômes (3.102) à (3.116) sont faibles devant la valeur de la surface, ce qui explique les normalisations en (3.101).

La matrice d'interaction liée aux invariants (3.101) peut être obtenue à partir de (3.60). On obtient alors (après des calculs fastidieux) :

$$\begin{aligned} \mathbf{L}_{r_i}^{\parallel} &= \begin{pmatrix} 0 & 0 & 0 & r_{i_{wx}} & r_{i_{wy}} & 0 \end{pmatrix} \\ \mathbf{L}_{r_j}^{\parallel} &= \begin{pmatrix} 0 & 0 & 0 & r_{j_{wx}} & r_{j_{wy}} & 0 \end{pmatrix} \end{aligned} \quad (3.117)$$

Les termes $r_{i_{wx}}$, $r_{i_{wy}}$, $r_{j_{wx}}$ et $r_{j_{wy}}$ ne sont pas donnés ici car ils n'ont pas une forme particulièrement sympathique. Comme prévu, on peut noter l'invariance de ces informations visuelles par rapport à tout mouvement de translation 3D (rappelons que nous considérons $A = B = 0$) ainsi que par rapport à ω_z , ils sont choisis tels que leurs matrices d'interaction associées $\mathbf{L}_{r_i}^{\parallel}$ et $\mathbf{L}_{r_j}^{\parallel}$ soient les plus orthogonales possible et présentent un domaine de convergence le plus large possible. En fait, ce choix dépend de la forme de l'objet par rapport auquel on veut se positionner. Nous étudierons dans le chapitre 5 ce choix pour deux sortes d'objets : une "baleine" et un "cerveau".

Enfin, dans le cas d'un objet symétrique (par exemple un rectangle), les moments centrés d'ordre impair deviennent nuls pour toutes les positions où l'image de l'objet est également symétrique. L'utilisation de la plupart de ces invariants pour contrôler les vitesses de rotation ω_x et ω_y est donc impossible. En ce qui concerne r_7 et r_8 qui dépendent des moments d'ordre 4, les résultats obtenus n'ont pas été satisfaisants. Pour ce genre d'objet, un bon découplage a été réalisé dans [Chaumette 02b], en utilisant les informations visuelles suivantes :

$$\begin{cases} s_x = (c_2 c_3 + s_2 s_3) / K \\ s_y = (s_2 c_3 - c_2 s_3) / K \end{cases} \quad (3.118)$$

où :

$$\begin{cases} c_2 = \mu_{03} - 3\mu_{21}, & s_2 = \mu_{30} - 3\mu_{12} \\ c_3 = c_1^2 - s_1^2, & s_3 = 2s_1 c_1 \\ K = I_1 I_2^{\frac{3}{2}} / \sqrt{a} \end{cases}$$

avec :

$$\begin{cases} c_1 = \mu_{20} - \mu_{02}, & s_1 = 2\mu_{11} \\ I_1 & = c_1^2 + s_1^2 \\ I_2 & = \mu_{20} + \mu_{02} \end{cases}$$

Dans le paragraphe suivant, nous traitons le choix des primitives dans le cas d'un objet discret.

3.6 Choix des informations visuelles dans le cas discret

Comme dans le cas continu, l'objectif pour le choix des informations visuelles est d'obtenir une matrice d'interaction creuse 6×6 et de rang plein et qui change

doucement autour de la position désirée. Nous considérons également comme dans le cas continu que la position désirée de la caméra est telle que son plan image soit parallèle à l'objet. Le cas d'une position désirée de la caméra où son plan image est non parallèle à l'objet sera traité dans la section suivante. Dans la section précédente, nous avons proposé les six informations visuelles suivantes pour contrôler les six degrés de liberté d'une caméra par rapport à un objet continu :

$$\mathbf{s} = (x_n, y_n, a_n, r_i, r_j, \alpha) \quad (3.119)$$

où:

$$\begin{cases} x_n = a_n x_g, & y_n = a_n y_g \\ a_n = Z^* \sqrt{\frac{a^*}{a}}, & \alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \end{cases}$$

$a(= m_{00})$ est égal à la surface de l'objet dans l'image, a^* est sa valeur désirée. Z^* est égale à la valeur désirée de la profondeur, α est l'orientation de l'objet dans l'image. r_i et r_j sont deux invariants aux rotations, aux translations et aux changements d'échelle donnés par (3.101). Dans le cas discret, toutes ces informations visuelles peuvent être utilisées, exceptée la surface m_{00} qui est égale au nombre de points dans le cas discret, et cela quelle que soit la position du robot. Nous proposons alors de remplacer la surface $a = m_{00}$ et sa valeur désirée a^* par :

$$a = \mu_{20} + \mu_{02} \text{ et } a^* = \mu_{20}^* + \mu_{02}^* \quad (3.120)$$

Ce choix aboutit à une matrice d'interaction similaire à celle associée à la surface dans le cas continu. En effet, la valeur de la matrice d'interaction \mathbf{L}_s^{\parallel} pour toutes les positions de la caméra telles que son plan image soit parallèle à l'objet a la forme suivante :

$$\mathbf{L}_s^{\parallel} = \begin{bmatrix} -1 & 0 & 0 & a_n \epsilon_{11} & -a_n(1 + \epsilon_{12}) & y_n \\ 0 & -1 & 0 & a_n(1 + \epsilon_{21}) & -a_n \epsilon_{11} & -x_n \\ 0 & 0 & -1 & -\epsilon_{31} & \epsilon_{32} & 0 \\ 0 & 0 & 0 & r_{i_{wx}} & r_{i_{wy}} & 0 \\ 0 & 0 & 0 & r_{j_{wx}} & r_{j_{wy}} & 0 \\ 0 & 0 & 0 & \alpha_{wx} & \alpha_{wy} & -1 \end{bmatrix} \quad (3.121)$$

avec :

$$\left\{ \begin{array}{l} \epsilon_{11} = n_{11} - x_g \epsilon_{31} \\ \epsilon_{12} = -(n_{20} - x_g \epsilon_{32}) \\ \epsilon_{21} = n_{02} - y_g \epsilon_{32} \\ \epsilon_{22} = -(n_{11} - y_g \epsilon_{31}) \\ \epsilon_{31} = y_g + (y_g \mu_{02} + x_g \mu_{11} + \mu_{21} + \mu_{03})/a \\ \epsilon_{32} = x_g + (x_g \mu_{20} + y_g \mu_{11} + \mu_{12} + \mu_{30})/a \\ \alpha_{wx} = (4\mu_{20}\mu_{12} + 2\mu_{20}x_g\mu_{02} + 2\mu_{20}y_g\mu_{11} \\ \quad - 4\mu_{02}\mu_{12} - 2x_g\mu_{02}^2 + 2\mu_{02}y_g\mu_{11} \\ \quad - 4x_g\mu_{11}^2 + 4\mu_{11}\mu_{03} - 4\mu_{11}\mu_{21})/d \\ \alpha_{wy} = (-4\mu_{20}\mu_{21} - 2y_g\mu_{20}^2 + 2\mu_{11}x_g\mu_{20} \\ \quad + 2\mu_{02}y_g\mu_{20} + 2\mu_{11}x_g\mu_{02} + 4\mu_{02}\mu_{21} \\ \quad - 4\mu_{11}\mu_{12} - 4y_g\mu_{11}^2 + 4\mu_{11}\mu_{30})/d \\ d = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \end{array} \right.$$

Comme dans le cas précédent, les termes $r_{i_{wx}}$, $r_{i_{wy}}$, $r_{j_{wx}}$ et $r_{j_{wy}}$ ont des formes analytiques trop complexes pour être données ici. Finalement, on peut noter que la matrice d'interaction associée aux informations visuelles choisies dans le cas discret est similaire à celle obtenue dans le cas continu. Des légères différences existent cependant sur les éléments non constants de cette matrice.

Finalement, dans tous les cas, on peut noter que \mathbf{L}_s^{\parallel} est triangulaire par bloc. Par ailleurs, si seulement un mouvement de translation doit être réalisé entre la position initiale de la caméra et sa position désirée, aucun mouvement de rotation ne sera généré et la caméra se déplacera sur une pure ligne droite avec les informations visuelles choisies, et ceci grâce au bloc diagonal constant égal à $-\mathbf{I}_3$ reliant les trois premières informations visuelles et les mouvements de translation. Toutefois, ce bon découplage n'est valable qu'autour d'une position désirée de la caméra telle que son plan image soit parallèle à l'objet. Dans la section suivante, nous généralisons ce découplage pour des positions de la caméra telles que le plan image soit non parallèle à l'objet.

3.7 Cas d'une position désirée de l'objet non parallèle au plan image

L'idée générale consiste à appliquer une rotation virtuelle à la caméra, à calculer la valeur des informations visuelles après ce mouvement virtuel, et à utiliser ces valeurs dans la boucle de commande. Plus précisément, la rotation est déterminée de façon à ce que le plan image dans sa position désirée soit parallèle à l'objet. Le découplage obtenu pour une position de l'objet parallèle au plan image peut alors être généralisé à n'importe quelle configuration de la caméra par rapport à un objet de forme plane.

La première étape de notre algorithme consiste à déterminer la rotation virtuelle \mathbf{R} à

appliquer à la caméra. Si la tâche est spécifiée par une pose désirée entre la caméra et l'objet, \mathbf{R} est directement donnée par cette pose désirée (mais cette méthode nécessite la connaissance du modèle de l'objet pour calculer l'image désirée). Si la tâche est spécifiée par une image désirée acquise pendant une étape d'apprentissage hors-ligne, la matrice de rotation \mathbf{R} doit être donnée également pendant cette étape, de la même manière qu'il faut spécifier la valeur de Z^* quand on considère la position parallèle. Ce procédé ne nécessite pas la connaissance du modèle 3D et une approximation grossière de \mathbf{R} suffit pour avoir de bonnes propriétés de découplage. Enfin, la valeur de cette matrice peut être déterminée en utilisant la méthode de calcul de pose que nous présentons dans le chapitre 4. Nous verrons que cette méthode ne nécessite que la connaissance d'une image de l'objet dans une position parallèle au plan image.

Notons (X_t, Y_t, Z_t) et (X, Y, Z) les coordonnées 3D d'un point après et avant le mouvement de rotation virtuelle. On a alors :

$$\begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.122)$$

À partir de cette équation, on peut déduire immédiatement les coordonnées (x_t, y_t) de la projection sur le plan image de la caméra du point après la rotation virtuelle. En effet, en utilisant l'équation de la projection perspective ($x_t = X_t/Z_t, y_t = Y_t/Z_t$), on obtient :

$$\begin{cases} x_t = (r_{11}X + r_{12}Y + r_{13}Z)/(r_{31}X + r_{32}Y + r_{33}Z) \\ y_t = (r_{21}X + r_{22}Y + r_{23}Z)/(r_{31}X + r_{32}Y + r_{33}Z) \end{cases} \quad (3.123)$$

En divisant par Z , on obtient :

$$\begin{cases} x_t = (r_{11}x + r_{12}y + r_{13})/(r_{31}x + r_{32}y + r_{33}) \\ y_t = (r_{21}x + r_{22}y + r_{23})/(r_{31}x + r_{32}y + r_{33}) \end{cases} \quad (3.124)$$

où (x, y) sont les coordonnées du point dans l'image réelle (donnée par $x = X/Z, y = Y/Z$). On peut noter que la transformation (3.124) peut être calculée directement à partir de \mathbf{R} et des coordonnées du point image. L'estimation du point 3D n'est donc pas nécessaire pour réaliser cette transformation. En combinant (3.124) et la définition des moments 2D dans le cas d'un objet continu, ces derniers peuvent être calculés après rotation par la formule suivante :

$$\begin{aligned} m_{t_{pq}} &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_t^p y_t^q dx_t dy_t \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left(\frac{r_{11}x + r_{12}y + r_{13}}{r_{31}x + r_{32}y + r_{33}} \right)^p \left(\frac{r_{21}x + r_{22}y + r_{23}}{r_{31}x + r_{32}y + r_{33}} \right)^q \det(\mathbf{J}_t) dx dy \end{aligned} \quad (3.125)$$

avec :

$$\mathbf{J}_t = \begin{pmatrix} \frac{\partial x_t}{\partial x} & \frac{\partial x_t}{\partial y} \\ \frac{\partial y_t}{\partial x} & \frac{\partial y_t}{\partial y} \end{pmatrix} = \frac{1}{(r_{31}x + r_{32}y + r_{33})^3} \quad (3.126)$$

3.7. CAS D'UNE POSITION DÉSIRÉE DE L'OBJET NON PARALLÈLE AU PLAN IMAGE 77

En combinant ce résultat avec (3.125), on obtient :

$$m_{t_{pq}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (r_{11}x + r_{12}y + r_{13})^p (r_{21}x + r_{22}y + r_{23})^q \frac{1}{(r_{31}x + r_{32}y + r_{33})^{p+q+3}} dx dy \quad (3.127)$$

Le calcul direct des moments sur l'image en utilisant l'équation (3.127) s'avère coûteux, surtout s'il doit être réalisé pour chaque image acquise dans le cas de l'asservissement visuel ou pour plusieurs itérations, par exemple dans le cas d'un calcul itératif de la pose comme nous le verrons dans le chapitre suivant. Pour réduire le coût de calcul des moments après un mouvement de rotation, on peut utiliser un développement de Taylor du terme $\frac{1}{(r_{31}x + r_{32}y + r_{33})^{p+q+3}}$. En effet, en faisant l'hypothèse que $r_{31}x + r_{32}y \ll r_{33}$, le développement de Taylor de ce terme peut s'écrire sous la forme suivante :

$$\frac{1}{(r_{31}x + r_{32}y + r_{33})^\gamma} \approx \frac{1}{r_{33}^\gamma} \left[1 - \gamma \frac{r_{31}x + r_{32}y}{r_{33}} + \gamma(\gamma + 1) \left(\frac{r_{31}x + r_{32}y}{r_{33}} \right)^2 - \dots \right] \quad (3.128)$$

où $\gamma = p + q + 3$. En remplaçant le terme $\frac{1}{(r_{31}x + r_{32}y + r_{33})^\gamma}$ par son développement de Taylor dans l'équation (3.127), on obtient alors :

$$m'_{pq} \approx \frac{1}{r_{33}^\gamma} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sum_{k_1=0}^p \sum_{l_1=0}^{k_1} \binom{k_1}{p} \binom{l_1}{k_1} \binom{k_2}{q} \binom{l_2}{k_2} \left(\frac{r_{11}}{r_{13}} \right)^{l_1} \left(\frac{r_{12}}{r_{13}} \right)^{k_1-l_1} \left(\frac{r_{21}}{r_{23}} \right)^{l_2} \left(\frac{r_{22}}{r_{23}} \right)^{k_2-l_2} x^{l_1+l_2} y^{k_1+k_2-l_1-l_2} \left(1 - \gamma \frac{(r_{31}x + r_{32}y)}{r_{33}} + \dots \right) dx dy \quad (3.129)$$

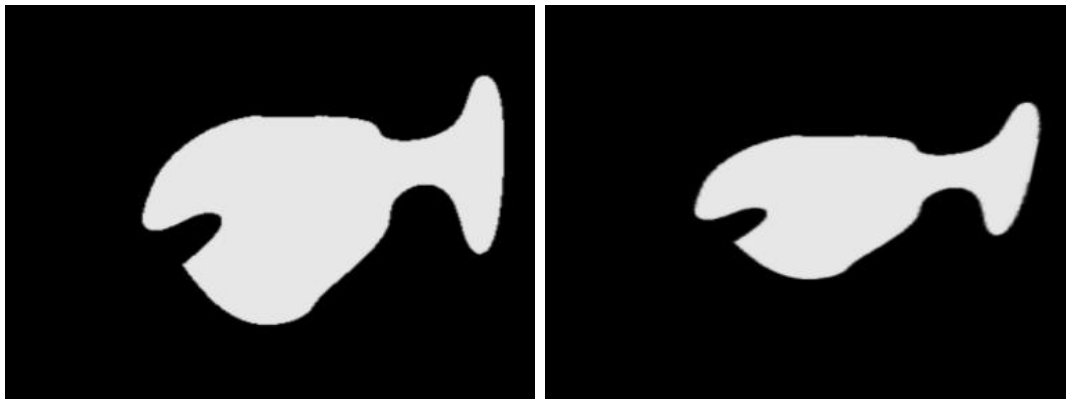
Après de simples développements, on obtient :

$$m'_{pq} \approx \frac{r_{13}^p r_{23}^q}{r_{33}^\gamma} \sum_{k_1=0}^p \sum_{l_1=0}^{k_1} \binom{k_1}{p} \binom{l_1}{k_1} \binom{k_2}{q} \binom{l_2}{k_2} \left(\frac{r_{11}}{r_{13}} \right)^{l_1} \left(\frac{r_{12}}{r_{13}} \right)^{k_1-l_1} \left(\frac{r_{21}}{r_{23}} \right)^{l_2} \left(\frac{r_{22}}{r_{23}} \right)^{k_2-l_2} (m_{l,k-l} - \frac{(r_{31}m_{l+1,k-l} + r_{32}m_{l,k-l+1})}{r_{33}} + \dots) \quad (3.130)$$

où $k = k_1 + k_2$, $l = l_1 + l_2$ et $m_{i,j}$ sont les moments calculés sur l'image courante. À partir des moments $m_{i,j}$ et de la matrice de rotation, on peut donc calculer une approximation des moments après rotation sur l'image. Le tableau 3.7 donne en pourcentage les erreurs sur les invariants r_9 , r_{10} et α obtenues à partir des deux images de la Figure 3.10 séparées par une rotation de 40° autour de l'axe x . D'après ce tableau, les erreurs sur ces valeurs deviennent négligeables à partir d'une approximation par développement de Taylor d'ordre 5.

Ordre du développement	$\frac{\Delta r_9 \times 100}{r_9}$	$\frac{\Delta r_{10} \times 100}{r_{10}}$	$\frac{\Delta \alpha \times 100}{\alpha}$
6	0.05 %	0.46 %	0.25 %
5	0.08 %	0.62 %	0.41 %
4	1.35 %	2.57 %	0.48 %
3	2.32 %	6.88 %	0.49 %

TAB. 3.1 – Erreur relative sur les invariants

FIG. 3.10 – Exemple de deux images d'une baleine où on a considéré une rotation de 40° selon l'axe X entre les deux positions de la caméra

Finalement, la commande est calculée à partir des moments transformés. La loi de commande est donnée alors par :

$$\mathbf{v} = -\lambda \mathbf{V} \widehat{\mathbf{L}}_{\mathbf{s}_t}^{-1} (\mathbf{s}_t - \mathbf{s}_t^*) \quad (3.131)$$

où \mathbf{s}_t et \mathbf{s}_t^* sont les valeurs des informations visuelles calculées respectivement pour la position courante et la position désirée de la caméra après le mouvement de rotation virtuelle. $\widehat{\mathbf{L}}_{\mathbf{s}_t}^{-1}$ est la valeur de la matrice de commande calculée à partir de ces informations (on rappelle que l'on utilise $\widehat{\mathbf{L}}_{\mathbf{s}} = \frac{1}{2}(\mathbf{L}_{\mathbf{s}(\mathbf{s}_t^*)}^{\parallel} + \mathbf{L}_{\mathbf{s}(\mathbf{s}_t)}^{\parallel})$). Finalement, la matrice \mathbf{V} représente le changement de repère du torseur cinématique entre la caméra virtuelle et la vraie caméra :

$$\mathbf{V} = \begin{bmatrix} \mathbf{R}^T & 0 \\ 0 & \mathbf{R}^T \end{bmatrix}$$

3.8 Conclusion

Dans ce chapitre, nous avons rappelé des notions sur la commande par asservissement visuel et en particulier la notion de matrice d'interaction. Par ailleurs, nous avons donné les formules analytiques de ces matrices dans le cas des moments. Nous avons présenté également les problèmes potentiels rencontrés en asservissement visuel. Pour remédier à ces problèmes, nous avons proposé six informations visuelles calculées à partir des moments invariants d'une image d'un objet plan. Ce choix d'informations visuelles a permis d'obtenir une matrice d'interaction creuse et de minimiser les non linéarités de celle-ci pour une position de la caméra où son plan image est parallèle à l'objet. Ce résultat a été généralisé également au cas d'une position de la caméra où son plan image n'est plus parallèle à l'objet. Dans le chapitre suivant, nous appliquons ces résultats au problème de calcul de pose par asservissement visuel virtuel.

Chapitre 4

Calcul de pose

Dans ce chapitre, nous traitons le problème du calcul de pose pour des objets plans de forme complexe. Comme nous l'avons vu dans le premier chapitre de ce document, ce problème est très classique en vision par ordinateur. Il consiste à calculer la transformation rigide ${}^c\mathbf{M}_o$ entre un repère lié à l'objet \mathcal{F}_o et le repère de la caméra \mathcal{F}_c dans une position inconnue. La relation entre les coordonnées d'un point de l'objet exprimées dans les deux repères \mathcal{F}_o et \mathcal{F}_c peut s'écrire :

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = {}^c\mathbf{M}_o \begin{pmatrix} X_o \\ Y_o \\ Z_o \\ 1 \end{pmatrix} \quad (4.1)$$

avec :

$${}^c\mathbf{M}_o = \begin{pmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ 0_{1 \times 3} & 1 \end{pmatrix} \quad (4.2)$$

où (X_o, Y_o, Z_o) et (X_c, Y_c, Z_c) sont les coordonnées d'un point de l'objet exprimées respectivement dans les repères \mathcal{F}_o et \mathcal{F}_c . ${}^c\mathbf{R}_o$ et ${}^c\mathbf{t}_o$ sont respectivement la matrice de rotation et le vecteur de translation entre les deux repères. Pour résoudre le problème du calcul de pose pour des objets de forme complexe, nous proposons dans ce chapitre deux méthodes de calcul en utilisant les moments invariants. La première méthode que nous développons est basée sur un asservissement visuel virtuel dont nous expliquerons le principe par la suite. La deuxième méthode, pour sa part, s'appuie sur un calcul exhaustif en utilisant une table de valeurs pour la pose. Les deux méthodes que nous proposons se basent sur le découplage que nous avons obtenu dans le chapitre précédent via un choix des informations visuelles. Ce découplage va être utilisé pour estimer dans une première étape la rotation entre les deux positions de la caméra. Une fois cette rotation annulée, la translation sera calculée en utilisant une formule analytique que nous donnons dans le paragraphe 4.2.4. Les deux méthodes que nous proposons nécessitent seulement la connaissance d'une image de l'objet pour une position telle que l'objet et le plan image de la caméra

soient parallèles. Ce qui revient à connaître la forme de l'objet plan à l'échelle près. Il est possible également d'appliquer les mêmes méthodes dans le cas où un modèle 3D de l'objet est disponible. Dans le paragraphe suivant, nous donnons un bref état de l'art sur les méthodes de calcul de pose à partir des moments.

4.1 État de l'art du calcul de pose en utilisant les moments

Dans la littérature, plusieurs méthodes ont été proposées pour résoudre le problème du calcul de pose à partir d'une image 3D de l'objet en utilisant les moments [Lo 89, Cyganski 85, Stokely 88]. Dans le cas où on ne dispose que d'une image obtenue à l'aide d'une caméra monoculaire, une méthode de calcul de pose a été proposée dans le cas où seulement un mouvement de rotation a été considéré entre les deux positions de la caméra [Mukundan 92, Coiffet 80]. Cette méthode utilise les moments invariants au mouvement de rotation autour de l'axe optique pour estimer le mouvement de rotation autour des axes x et y de la caméra. La valeur de cette rotation est trouvée à partir d'une table de valeur de pose. Dans [Mukundan 96], une méthode est proposée qui considère une transformation générale composée de mouvements de translation et de rotation dans le cas d'un objet plan. Cependant, cette méthode considère qu'un mouvement général composé de translation et de rotation de la caméra est équivalent à une transformation affine dans l'image. Cela est exact dans le cas d'une projection orthographique mais ne l'est plus pour une projection perspective. En effet, si on considère le cas d'un mouvement de la caméra donné par :

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (4.3)$$

en utilisant les équations de la projection perspective (3.21) et l'équation du plan (3.49) dans (4.3), la transformation dans l'image peut s'écrire :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \frac{1}{r_{31}x + r_{32}y + r_{33} + \frac{t_z}{Z}} \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \frac{1}{Z(r_{31}x + r_{32}y + r_{33}) + t_z} \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (4.4)$$

où $Z = \frac{1}{Ax + By + C}$. Dans [Mukundan 96], les termes $\frac{1}{r_{31}x + r_{32}y + r_{33} + \frac{t_z}{Z}}$ et $\frac{1}{Z(r_{31}x + r_{32}y + r_{33}) + t_z}$ ont été supposés constants. La transformation considérée dans l'image est de la forme :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \gamma_1 \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \gamma_2 \begin{pmatrix} i_x \\ i_y \end{pmatrix} \quad (4.5)$$

Cette transformation n'est valide que dans le cas où la caméra peut être modélisée par une projection orthographique (c'est-à-dire que l'on suppose une profondeur constante pour tous les points de l'objet).

4.2 Pose par asservissement visuel virtuel

Dans ce paragraphe, nous élaborons une méthode de calcul de pose en se basant sur un asservissement visuel virtuel. Nous expliquerons dans la suite le principe d'un asservissement visuel virtuel. Puis nous appliquons ce formalisme pour calculer la pose entre deux positions de la caméra.

4.2.1 Asservissement visuel virtuel

Le principe de l'asservissement visuel virtuel (AVV) est similaire à celui d'un asservissement visuel classique (AV) [Marchand 02]. La seule différence entre les deux types d'asservissement est que les informations visuelles sont extraites d'une image acquise dans le cas d'un AV alors qu'elles sont calculées à partir d'un modèle et d'une position d'une caméra virtuelle dans le cas d'un AVV. En fait, l'AVV consiste à déplacer une caméra virtuelle d'une position initiale connue par une transformation rigide entre le repère lié à l'objet \mathcal{F}_o et le repère caméra \mathcal{F}_i (voir figure 4.1) vers une position finale inconnue qu'on doit déterminer en minimisant une fonction de tache comme en AV. La position finale après convergence correspond à la pose recherchée. Le déplacement de la caméra virtuelle est calculé en utilisant la même loi de commande utilisée dans le cas d'un AV. Ainsi, la fonction de tache est calculée à partir de l'erreur sur les primitives calculées pour la position de la caméra virtuelle et celles calculées sur l'image pour laquelle la pose doit être calculée.

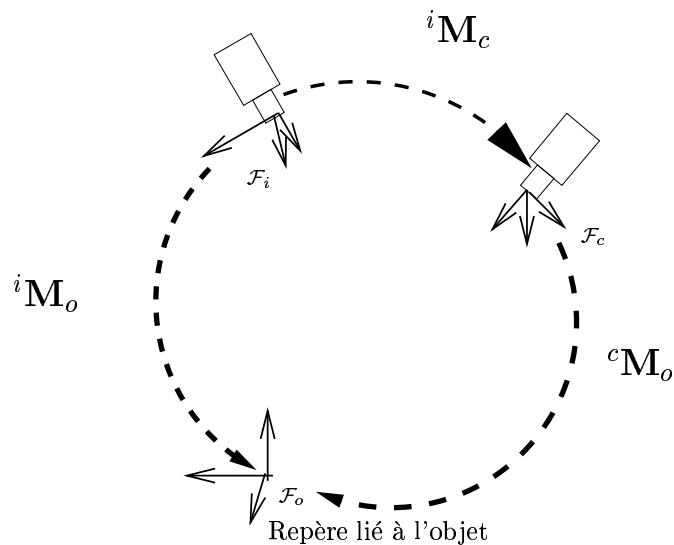


FIG. 4.1 – *Principe d'un asservissement visuel virtuel*

4.2.2 Principe de la méthode proposée

Dans le chapitre précédent, nous avons proposé six informations visuelles basées sur les moments pour contrôler les six degrés de liberté de la caméra :

$$\mathbf{s} = (x_n, y_n, a_n, r_i, r_j, \alpha) \quad (4.6)$$

On rappelle que la matrice d'interaction associée à ces six informations visuelles pour une position de la caméra telle que son plan image soit parallèle à l'objet a la forme suivante :

$$\begin{aligned} \mathbf{L}_{x_n}^{\parallel} &= (-1 \ 0 \ 0 \ a_n \epsilon_{11} \ -a_n(1+\epsilon_{12}) \ y_n) \\ \mathbf{L}_{y_n}^{\parallel} &= (0 \ -1 \ 0 \ a_n(1+\epsilon_{21}) \ -a_n \epsilon_{11} \ -x_n) \\ \mathbf{L}_{a_n}^{\parallel} &= (0 \ 0 \ -1 \ -3y_n/2 \ 3x_n/2 \ 0) \\ \mathbf{L}_{r_i}^{\parallel} &= (0 \ 0 \ 0 \ r_{iwx} \ r_{iwy} \ 0) \\ \mathbf{L}_{r_j}^{\parallel} &= (0 \ 0 \ 0 \ r_{jwx} \ r_{jwy} \ 0) \\ \mathbf{L}_{\alpha}^{\parallel} &= (0 \ 0 \ 0 \ \alpha_{wx} \ \alpha_{wy} \ -1) \end{aligned} \quad (4.7)$$

On rappelle de nouveau le bon découplage entre les degrés de liberté de la caméra en utilisant ces combinaisons de moments. Pour cette raison, ces informations visuelles seront utilisées pour le calcul de pose étant donnée la dualité entre le calcul de pose et l'asservissement visuel. Cependant, la méthode qu'on propose est légèrement différente d'un AVV dont nous avons expliqué le principe précédemment. En effet, pour garder le bon découplage obtenu pour une pose désirée où le plan image est parallèle à l'objet, nous considérons que la position désirée de la caméra est une position qu'on connaît et telle que le plan image soit parallèle à l'objet. Ce découplage assurera que la matrice d'interaction ne soit pas singulière pour la position désirée et fournit un large domaine de convergence. On note cette pose désirée ${}^p\mathbf{M}_o$ (voir figure 4.2).

La pose ${}^c\mathbf{M}_o$ qu'on doit déterminer sera obtenue à partir du déplacement ${}^c\mathbf{M}_p$ de la caméra virtuelle réalisé par AVV pour atteindre cette pose désirée en partant de la pose inconnue. Plus précisément, on a bien sûr :

$${}^c\mathbf{M}_o = {}^c\mathbf{M}_p {}^p\mathbf{M}_o \quad (4.8)$$

Cependant, seulement des mouvements de rotation peuvent être considérés dans ce cas dans la boucle de commande. En effet, le calcul des informations visuelles après un mouvement de translation nécessite la connaissance de l'orientation de l'objet par rapport à la caméra. Plus précisément, les coordonnées d'un point dans l'image après un mouvement de translation de la caméra peuvent s'écrire :

$$\begin{cases} x' = \frac{X'}{Z'} = \frac{X+t_x}{Z+t_z} = \frac{x}{1+\frac{t_z}{Z}} + \frac{t_x}{Z+t_z} \\ y' = \frac{Y'}{Z'} = \frac{Y+t_y}{Z+t_z} = \frac{y}{1+\frac{t_z}{Z}} + \frac{t_y}{Z+t_z} \end{cases} \quad (4.9)$$

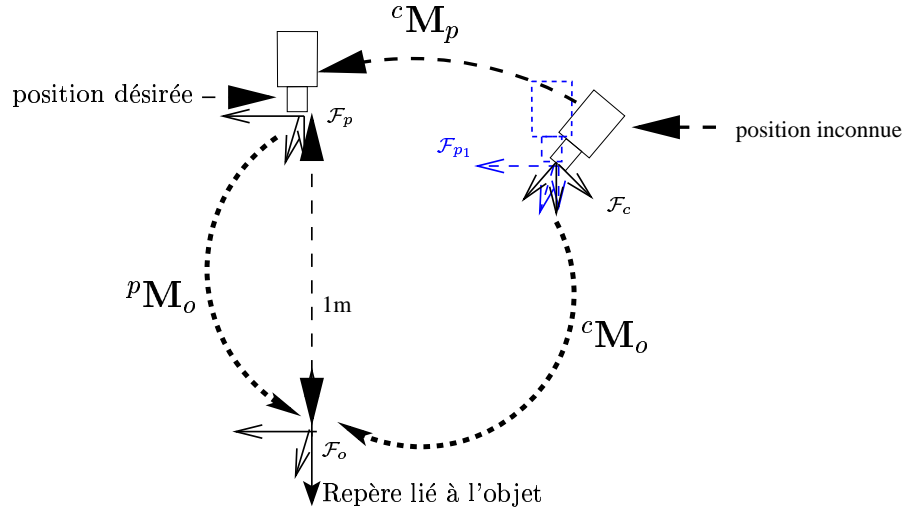


FIG. 4.2 – Principe du calcul de rotation

Or $\frac{1}{Z} = Ax + By + C$, où A , B et C sont les paramètres qui définissent le plan de l'objet dans le repère caméra. Ceux-ci ne sont pas disponibles tant que le calcul de pose n'est pas achevé. Nous calculons donc la pose en deux étapes. D'abord, on estime la rotation en utilisant un AVV où seulement des mouvements de rotation sont considérés, puis on calcule la translation en annulant l'erreur entre les informations visuelles (x_n, y_n, a_n) .

4.2.3 Calcul de la rotation

Nous considérons comme position désirée celle référencée par le repère \mathcal{F}_p (voir figure 4.2) où la transformation ${}^p\mathbf{M}_o$ est donnée par :

$${}^p\mathbf{M}_o = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.10)$$

Les moments de l'image de l'objet pour cette position peuvent être calculés directement à partir du modèle de l'objet. En effet, en utilisant (4.10), on a :

$$\begin{cases} X_c = X_o \\ Y_c = Y_o \\ Z_c = Z_o + 1 = 1 \end{cases} \quad (4.11)$$

En combinant ce résultat avec les équations de la projection perspective (3.21), on obtient :

$$\begin{cases} x_c = X_o \\ y_c = Y_o \end{cases} \quad (4.12)$$

Finalement, en utilisant (4.12) dans la définition des moments 2D, on en déduit que :

$$m_{ij} = M_{ij} \quad (4.13)$$

où M_{ij} sont les moments calculés à partir du modèle de l'objet. Considérons maintenant un repère \mathcal{F}_{p_1} de même orientation que \mathcal{F}_p et dont le centre coïncide avec celui de \mathcal{F}_c . Alors, la transformation entre les repères \mathcal{F}_p et \mathcal{F}_{p_1} est donnée par :

$${}^{p_1}\mathbf{M}_p = \begin{pmatrix} I_3 & {}^{p_1}\mathbf{t}_p \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix} \quad (4.14)$$

Les informations visuelles r_i , r_j et α auront les mêmes valeurs pour les deux positions de la caméra référencées par les repères \mathcal{F}_p et \mathcal{F}_{p_1} , puisque le mouvement considéré entre ces deux repères est une translation pure et puisque r_i , r_j et α sont invariants aux translations 3D quand l'objet est parallèle au plan image. En annulant l'erreur entre les trois informations visuelles r_i , r_j et α calculées pour la position désirée et la position à déterminer, on détermine ainsi le mouvement de rotation entre les deux repères \mathcal{F}_{p_1} et \mathcal{F}_c (sauf existence d'un minimum local). Le mouvement de la caméra est calculé en utilisant une loi classique d'asservissement visuel mais en ne considérant que des mouvements de rotation. Plus précisément, on a :

$$\boldsymbol{\omega} = -\lambda \left(\frac{\mathbf{L}_\omega(\mathbf{s}) + \mathbf{L}_\omega(\mathbf{s}^*)}{2} \right)^{-1} (\mathbf{s} - \mathbf{s}^*)$$

où :

$$\mathbf{L}_\omega = \begin{pmatrix} r_{iwx} & r_{iwy} & 0 \\ r_{jwx} & r_{jwy} & 0 \\ \alpha_{wx} & \alpha_{wy} & -1 \end{pmatrix}$$

Les moments après chaque mouvement de rotation sont calculés en utilisant l'équation (3.130) pour réduire le temps de calcul. Finalement, un résumé de notre algorithme est donné par le schéma 4.3.

Phase d'initialisation– **calcul des informations visuelles**donner la valeur du gain scalaire λ calcul des moments pour la position désirée m_{ij}^* calcul de $\mathbf{s}^* = (r_i^*, r_j^*, \alpha^*)$ calcul des moments pour la position inconnue m_{ij} calcul de $\mathbf{s} = (r_i, r_j, \alpha)$ – **initialisation de la matrice de rotation**

$${}^{c_1}\mathbf{R}_p(1) = \mathbf{R}_{(\alpha^* - \alpha)}$$

tant que $er = (r_i^* - r_i)^2 + (r_j^* - r_j)^2 + (\alpha^* - \alpha)^2 > \epsilon$ – **calcul du torseur cinématique**

$$\boldsymbol{\omega} = -\lambda \mathbf{L}_\omega^{-1}(\mathbf{s} - \mathbf{s}^*)$$

avec

$$\mathbf{L}_\omega = \begin{pmatrix} r_{iwx} & r_{iwy} & 0 \\ r_{jwx} & r_{jwy} & 0 \\ \alpha_{wx} & \alpha_{wy} & -1 \end{pmatrix}$$

– **calcul de la matrice de rotation $\Delta\mathbf{R}_\theta = {}^{c_{k+1}}\mathbf{R}_{c_k}$**

$$\theta = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$$

$$\mathbf{u}_\theta = \frac{\boldsymbol{\omega}}{\theta}$$

$$\Delta\mathbf{R}_\theta = \mathbb{I}_3 + \sin \theta [\mathbf{u}]_\times + (1 - \cos \theta) [\mathbf{u}]_\times^2$$

avec

$$[\mathbf{u}]_\times = \begin{pmatrix} 0 & -u(3) & u(2) \\ u(3) & 0 & -u(1) \\ -u(2) & u(1) & 0 \end{pmatrix}$$

et

$$[\mathbf{u}]_\times^2 = \begin{pmatrix} u(1)^2 - 1 & u(1)u(2) & u(1)u(3) \\ u(1)u(2) & u(2)^2 - 1 & u(2)u(3) \\ u(1)u(3) & u(2)u(3) & u(3)^2 - 1 \end{pmatrix}$$

$${}^{c_{k+1}}\mathbf{R}_p = \Delta\mathbf{R}_\theta {}^{c_k}\mathbf{R}_p$$

calcul des nouvelles valeurs des moments m_{ij} à partir de leurs valeurs initiales et ${}^{c_{k+1}}\mathbf{R}_p$ en utilisant l'équation (3.130)calcul de \mathbf{s} à partir des nouvelles valeurs de m_{ij}

FIG. 4.3 – Méthode de calcul de rotation par AVV

4.2.4 Calcul de la translation

Dans ce paragraphe, nous considérons que le mouvement de rotation entre les deux positions de la caméra référencées par les repères \mathcal{F}_p et \mathcal{F}_c a été estimée en utilisant la méthode que nous avons proposée dans le paragraphe précédent. Dans ce cas, le repère \mathcal{F}_c après rotation coïncide avec le repère \mathcal{F}_{p_1} . Le mouvement de translation entre les repères \mathcal{F}_{p_1} et \mathcal{F}_p peut être ensuite calculé par la relation suivante :

$$\begin{cases} t_x &= x_n - x_n^* \\ t_y &= y_n - y_n^* \\ t_z &= a_n - a_n^* \end{cases} \quad (4.15)$$

où (x_n^*, y_n^*, a_n^*) et (x_n, y_n, a_n) sont calculés respectivement pour la position donnée par les repères \mathcal{F}_p et \mathcal{F}_{p_1} . Plus précisément, pour (x_n, y_n, a_n) , on utilise (3.130) en considérant l'image correspondant à \mathcal{F}_c et la rotation qu'on vient de déterminer. En effet, le bloc de la matrice d'interaction associée à ces trois informations visuelles et qui correspond aux mouvements de translations est égal à $-I_3$ pour n'importe quelle position de la caméra où le plan image est parallèle à l'objet. On peut facilement en déduire que la translation entre \mathcal{F}_p et \mathcal{F}_{p_1} peut être calculée par (4.15). Dans la suite, nous validons notre approche sur des images de synthèse.

4.2.5 Résultats

Dans ce paragraphe, on teste la méthode de calcul de pose en utilisant un AVV pour des déplacements comportant des grands mouvements de translation et de rotation. Les premières images utilisées sont données sur la figure 4.4.

La matrice de rotation a été initialisée par la valeur de la matrice correspondant à une rotation d'un angle égal à $\alpha^* - \alpha$ autour de l'axe Z de la caméra entre les deux positions référencées par les repères \mathcal{F}_p et \mathcal{F}_c . Cette initialisation permet quasiment d'annuler la rotation autour de l'axe optique de la caméra. Cependant, elle ne change pas l'orientation du plan image par rapport à l'objet. La figure 4.5 donne la décroissance du module de l'erreur sur les informations visuelles r_9 , r_{10} et α avec un gain $\lambda = 0.2$. Ces courbes prouvent la convergence des valeurs des informations visuelles vers leurs valeurs désirées d'une façon exponentielle sauf pour la première itération où on observe une chute brutale due à l'initialisation de la pose (puisque l'on annule $\alpha^* - \alpha$). Les valeurs de la transformation ${}^c\mathbf{M}_p$ et ses estimées en utilisant la méthode proposée pour chaque image de la figure 4.4 sont les suivantes :

$$\mathbf{M}_1 = \begin{pmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 0.7663 & 0.6425 & -0.6425 \\ 0 & -0.6425 & 0.7663 & 0.2337 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \quad \widehat{\mathbf{M}}_1 = \begin{pmatrix} 0.9999 & -0.0036 & -0.0068 & 0.0074 \\ 0.0071 & 0.7615 & 0.6481 & -0.6474 \\ 0.0028 & -0.6481 & 0.7615 & 0.2385 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{U}\theta = (-40 \ 0 \ 0)^o \quad \widehat{\mathbf{U}}\theta = (-40.4215 \ -0.3012 \ 0.3381)^o \quad (4.16)$$

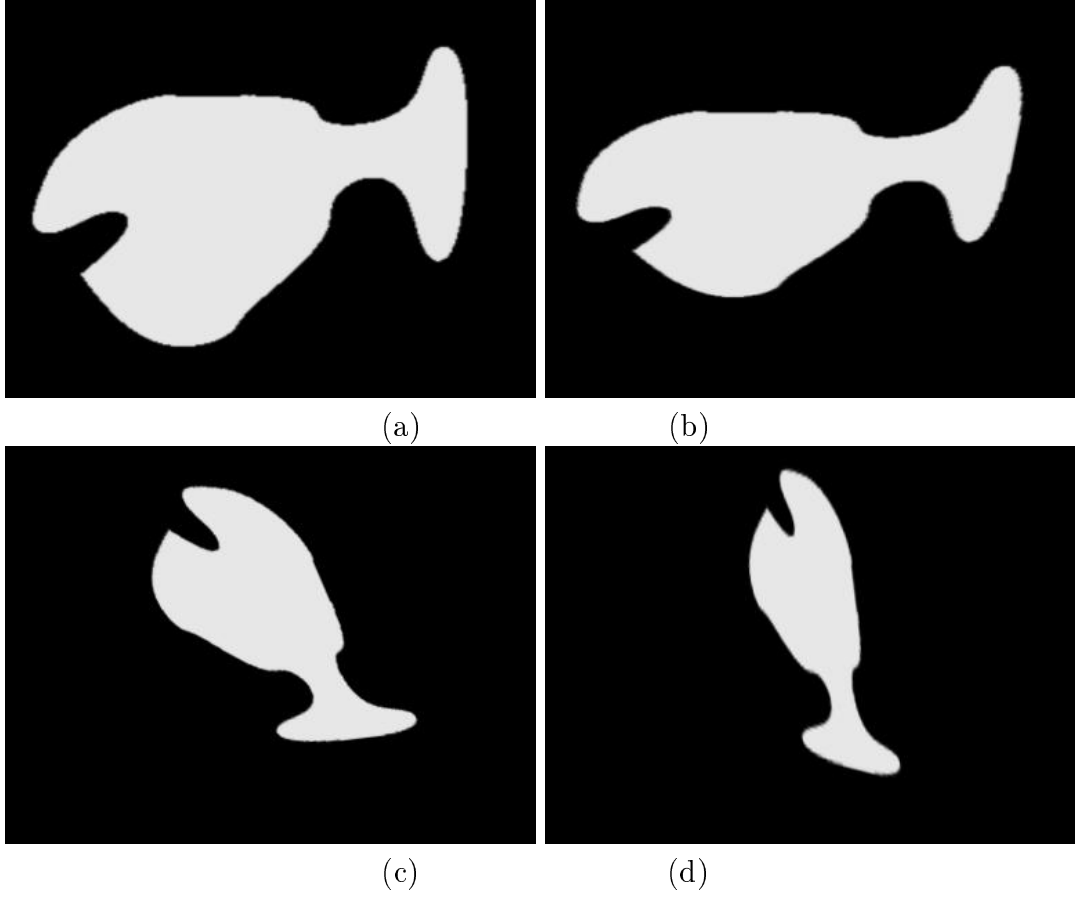


FIG. 4.4 – Images utilisées pour le calcul de pose : (a) image obtenue pour la position parallèle, (b) image obtenue après la transformation \mathbf{M}_1 , (c) image obtenue après la transformation \mathbf{M}_2 , (d) image obtenue après la transformation \mathbf{M}_3

$$\begin{aligned}
 \mathbf{M}_2 &= \begin{pmatrix} 0.3808 & 0.8528 & 0.3575 & -0.5006 \\ -0.8528 & 0.1744 & 0.4924 & -0.6893 \\ 0.3575 & -0.4924 & 0.7936 & -0.1110 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} & \widehat{\mathbf{M}}_2 &= \begin{pmatrix} 0.3804 & 0.8519 & 0.3598 & -0.5048 \\ -0.8545 & 0.1749 & 0.4890 & -0.6886 \\ 0.3536 & -0.4935 & 0.7945 & -0.1170 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 \mathbf{U}\theta &= (-40 \quad 0 \quad -69)^\circ & \widehat{\mathbf{U}}\theta &= (-39.9017 \quad 0.2545 \quad -69.2974)^\circ
 \end{aligned} \tag{4.17}$$

$$\begin{aligned}
 \mathbf{M}_3 &= \begin{pmatrix} 0.1204 & 0.9918 & 0.1055 & -0.1414 \\ -0.5655 & -0.0178 & 0.8295 & -1.1494 \\ 0.8216 & -0.1554 & 0.5602 & 0.2296 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \widehat{\mathbf{M}}_3 &= \begin{pmatrix} 0.1244 & 0.9855 & 0.1149 & -0.1606 \\ -0.5667 & -0.0244 & 0.8235 & -1.1636 \\ 0.8144 & -0.1675 & 0.5554 & 0.2163 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 \mathbf{U}\theta &= (-50 \quad -35.36 \quad -79.06)^\circ & \widehat{\mathbf{U}}\theta &= (-50.2949 \quad -35.4981 \quad -78.7689)^\circ
 \end{aligned} \tag{4.18}$$

Ces équations montrent à leur tour que la pose a convergé vers sa valeur désirée dans chacun des cas traités. Ces résultats prouvent également que le domaine de convergence de la méthode utilisée est très large étant donné que le mouvement de rotation autour des axes X et Y entre la position initiale et la position désirée est très grand. Les erreurs entre les vraies valeurs et les valeurs estimées sont dues aux erreurs de l'approximation par un développement de Taylor pour calculer les moments après un mouvement de rotation.

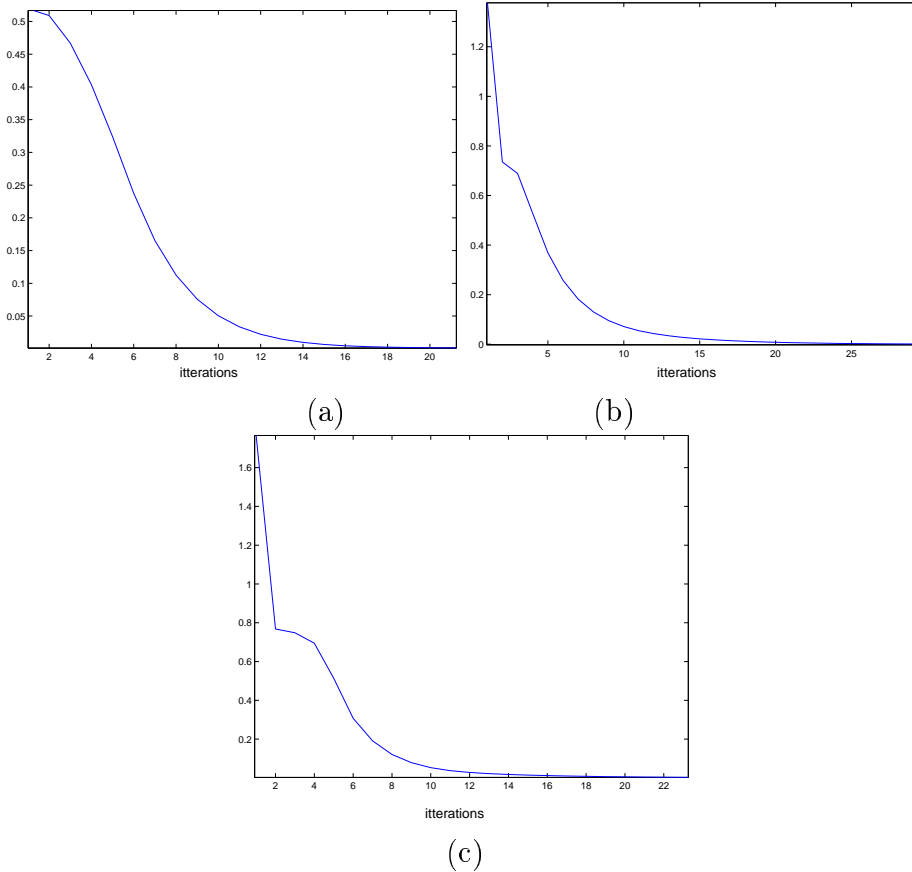


FIG. 4.5 – *Décroissance du module de l'erreur*: (a) résultat obtenu pour la transformation \mathbf{M}_1 , (b) résultat obtenu pour la transformation \mathbf{M}_2 , (c) résultat obtenu pour la transformation \mathbf{M}_3

Par contre, dans le cas de l'image donnée sur la figure 4.6 correspondant à la transformation suivante :

$$\mathbf{M}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.2595 & 0.9658 & -1.1589 \\ 0 & -0.9658 & 0.2595 & 0.6886 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \quad \widehat{\mathbf{M}}_4 = \begin{pmatrix} 0.8686 & 0.0858 & -0.4878 & 0.6529 \\ 0.3982 & 0.4645 & 0.7909 & -0.9218 \\ 0.2945 & -0.8813 & 0.3693 & 0.5795 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{U}\theta = (-75 \ 0 \ 0)^o \quad \widehat{\mathbf{U}}\theta = (-62.03 \ -29.02 \ 11.59)^o$$

(4.19)

on peut noter que le système converge vers un minimums local, en raison du très grand déplacement considéré (une rotation de 75° par rapport à l'axe x de la caméra). Pour éviter ce problème, nous proposons dans la section suivante une méthode basée sur un calcul d'une table d'erreur pour initialiser correctement la méthode utilisant un AVV.

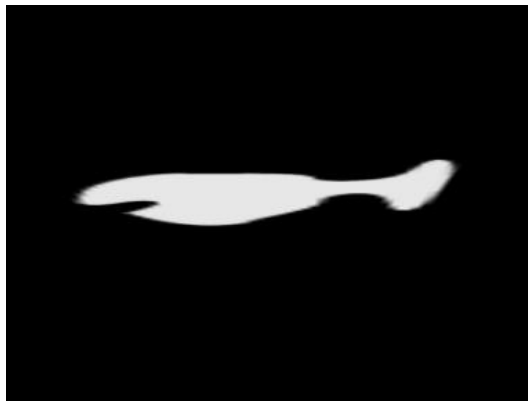


FIG. 4.6 – Image obtenue après la transformation M_4

4.3 Pose en utilisant une table de valeur

Dans ce paragraphe, nous proposons une méthode simple de calcul de pose basée sur un calcul exhaustif. Nous comparons ensuite cette méthode à celle proposée dans le paragraphe précédent.

4.3.1 Principe de la méthode

De la même manière que pour la méthode précédente, on estime en deux étapes la valeur de la pose, tout d'abord en déterminant la rotation entre les deux positions de la caméra puis en calculant la translation en utilisant l'équation (4.15).

Supposons que la position inconnue de la caméra est telle que le plan image soit parallèle à l'objet. Dans ce cas de figure, il suffit d'appliquer une rotation de $\alpha^* - \alpha$ autour de l'axe Z de la caméra pour annuler la rotation entre le repère lié à la caméra dans sa position inconnue et le repère lié à la caméra dans sa position désirée. Pour se ramener à une position de la caméra telle que son plan image soit parallèle à l'objet, il suffit de calculer la rotation autour d'un axe appartenant au plan défini par les deux axes X et Y qui annule les erreurs sur les invariants r_i donnés par (3.101) (deux ou plus de ces invariants sont utilisés). Nous verrons dans le chapitre suivant comment choisir les combinaisons d'invariants pour calculer l'erreur e_r .

Pour calculer le mouvement de rotation, on construit un tableau du module d'erreur $e_r = \sum_i (r_i - r_i^*)^2$ en appliquant des rotations comprises entre des valeurs minimale et maximale avec un pas donné autour de l'axe X et de l'axe Y . On

cherchera dans ce tableau la valeur de la rotation qui minimise l'erreur e_r . Pour calculer la valeur des r_i pour les différentes rotations à appliquer, on utilise la même méthode basée sur le développement de Taylor (3.130). Le calcul de la table d'erreur est détaillé sur le schéma 4.7.

Phase d'initialisation

- calcul des moments $m_{c_{ij}}$ pour l'image acquise à la position à calculer
- calcul des moments $m_{p_{ij}}$ pour l'image désirée (c'est-à-dire celle qui correspond à la position parallèle)
- calcul de r_i^* à partir de $m_{p_{ij}}$
- introduire le pas $\Delta\theta$ et la taille du domaine de recherche.

calcul de e_r pour un élément (i, j) de la table

- vecteur de rotation :

$$\mathbf{u} = (1 \quad 0 \quad 0) \text{ si } i = j = 0$$

$$\mathbf{u} = \left(\frac{i}{\sqrt{i^2+j^2}} \quad \frac{j}{\sqrt{i^2+j^2}} \quad 0 \right) \text{ si non}$$

- angle de rotation $\theta = \sqrt{i^2 + j^2} \Delta\theta$:

- matrice de rotation $\mathbf{R} = \mathbf{I}_3 + \sin \theta [\mathbf{u}]_{\times} + (1 - \cos \theta) [\mathbf{u}]_{\times}^2$
avec

$$[\mathbf{u}]_{\times} = \begin{pmatrix} 0 & -u(3) & u(2) \\ u(3) & 0 & -u(1) \\ -u(2) & u(1) & 0 \end{pmatrix}$$

et

$$[\mathbf{u}]_{\times}^2 = \begin{pmatrix} u(1)^2 - 1 & u(1)u(2) & u(1)u(3) \\ u(1)u(2) & u(2)^2 - 1 & u(2)u(3) \\ u(1)u(3) & u(2)u(3) & u(3)^2 - 1 \end{pmatrix}$$

- calcul des valeurs des moments m_{ij} après la rotation \mathbf{R} à partir de $m_{c_{ij}}$ en utilisant l'équation (3.130)
- calcul des informations visuelles r_i à partir de m_{ij} .
- calcul de $e_r = \sum_i (r_i - r_i^*)^2$

remarque : le calcul de chaque élément de la table d'erreur ne dépend pas des autres éléments de cette table.

FIG. 4.7 – Construction de la table d'erreurs

4.3.2 Résultats

Nous traitons les mêmes cas de figure étudiés en utilisant la méthode du calcul de pose par AVV (c'est-à-dire les cas donnés par les figures 4.4). Les valeurs des transformations ${}^c\mathbf{M}_p$ entre les deux positions de la caméra et leurs estimées sont données ci-dessous :

$$\mathbf{M}_1 = \begin{pmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 0.7663 & 0.6425 & -0.6425 \\ 0 & -0.6425 & 0.7663 & 0.2337 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \quad \widehat{\mathbf{M}}_1 = \begin{pmatrix} 1 & 0 & 0 & 0.0003 \\ 0 & 0.7660 & 0.6427 & -0.643 \\ 0 & -0.6427 & 0.7660 & 0.2312 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{U}\theta = (-40 \ 0 \ 0)^\circ \quad \widehat{\mathbf{U}}\theta = (-40.0 \ 0 \ 0)^\circ \quad (4.20)$$

$$\mathbf{M}_2 = \begin{pmatrix} 0.3808 & 0.8528 & 0.3575 & -0.5006 \\ -0.8528 & 0.1744 & 0.4924 & -0.6893 \\ 0.3575 & -0.4924 & 0.7936 & -0.1110 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \quad \widehat{\mathbf{M}}_2 = \begin{pmatrix} 0.3838 & 0.8542 & 0.3506 & -0.4918 \\ -0.8513 & 0.1803 & 0.4925 & -0.6941 \\ 0.3575 & -0.4875 & 0.7965 & -0.1206 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{U}\theta = (-40.00 \ 0 \ -69.28)^\circ \quad \widehat{\mathbf{U}}\theta = (-39.68 \ -0.27 \ -69.05)^\circ \quad (4.21)$$

$$\mathbf{M}_3 = \begin{pmatrix} 0.1204 & 0.9918 & 0.1055 & -0.1414 \\ -0.5655 & -0.0178 & 0.8295 & -1.1494 \\ 0.8216 & -0.1554 & 0.5602 & 0.2296 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \widehat{\mathbf{M}}_3 = \begin{pmatrix} 0.1242 & 0.9865 & 0.1065 & -0.1485 \\ -0.5677 & -0.0173 & 0.8230 & -1.1650 \\ 0.8137 & -0.1627 & 0.5578 & 0.2115 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{U}\theta = (-49.83 \ -36.23 \ -78.80)^\circ \quad \widehat{\mathbf{U}}\theta = (-49.84 \ -35.76 \ -78.59)^\circ \quad (4.22)$$

$$\mathbf{M}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.2595 & 0.9658 & -1.1589 \\ 0 & -0.9658 & 0.2595 & 0.6886 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \quad \widehat{\mathbf{M}}_4 = \begin{pmatrix} 0.9999 & -0.0011 & -0.0044 & 0.0034 \\ 0.0046 & 0.2588 & 0.9659 & -1.2048 \\ 0 & -0.9659 & 0.2588 & 0.6753 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{U}\theta = (-75.00 \ 0 \ 0)^\circ \quad \widehat{\mathbf{U}}\theta = (-75.03 \ -0.17 \ 0.22)^\circ \quad (4.23)$$

On peut noter qu'une bonne valeur de la pose a été obtenue même pour l'image 4.6. La figure 4.8 donne la carte de $f(e_r) = \frac{1}{cste + e_r}$ ($f(e_r)$ est maximal quand e_r est minimale) en fonction des rotations appliquées virtuellement sur la caméra dans sa position inconnue selon les deux axes X et Y . Nous avons choisi la représentation $f(e_r)$ au lieu de e_r pour mettre en évidence les pics correspondant aux minima. À partir de ces figures, on peut noter l'existence d'un minimum local (petit pic) quasiment symétrique au minimum global (grand pic). On retrouve ainsi le problème mentionné dans [Oberkampf 96] pour un ensemble de points coplanaires. Nous traitons d'ailleurs ce cas de figure en utilisant les moments dans la section suivante.

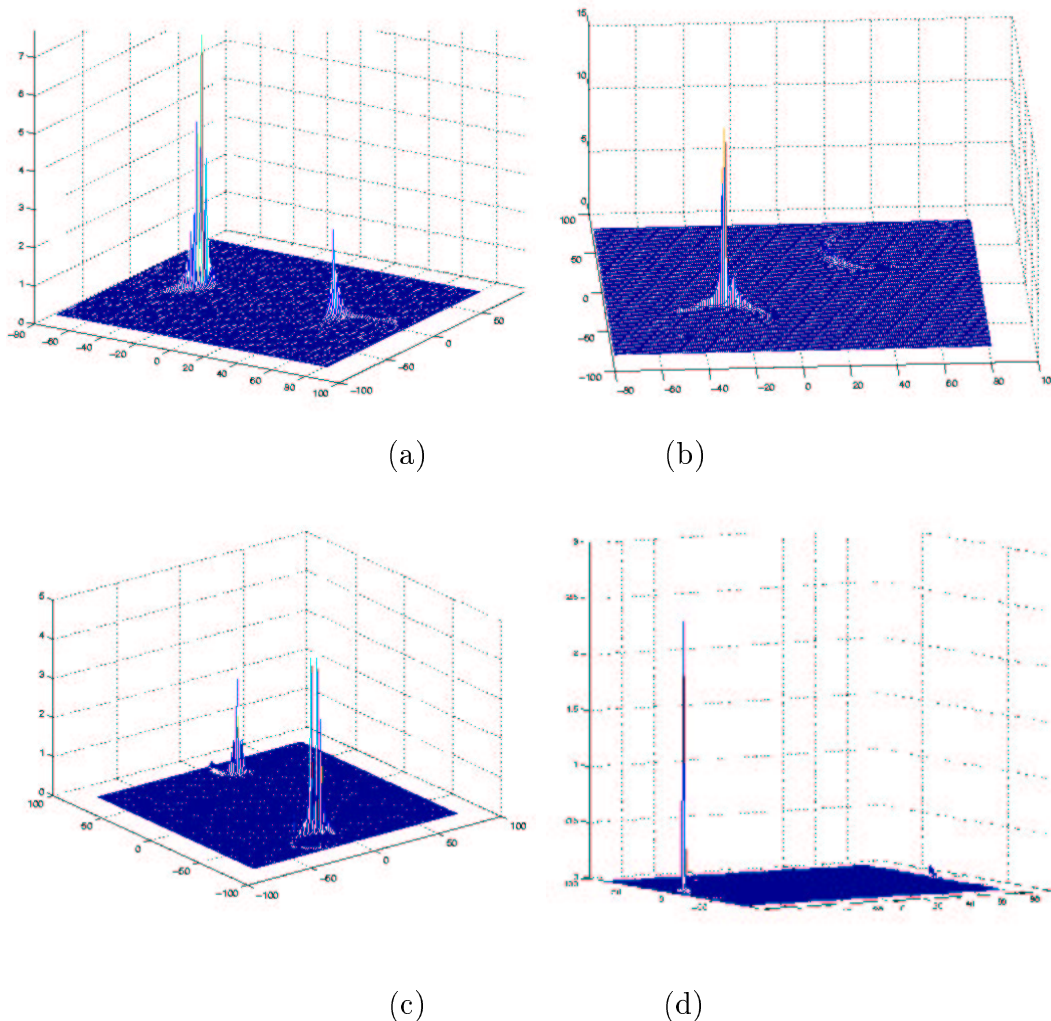


FIG. 4.8 – Amplitude de $f(e_r) = \frac{1}{cste + e_r}$ en fonction de la rotation: (a) résultat obtenu pour \mathbf{M}_1 , (b) résultat obtenu pour \mathbf{M}_2 , (c) résultat obtenu pour \mathbf{M}_3 , (d) résultat obtenu pour \mathbf{M}_4

4.4 Application au cas discret

Dans la section précédente, nous avons proposé deux méthodes de calcul de pose dans le cas d'objets continus de forme quelconque. La première méthode est basée sur un calcul itératif de pose par AVV et présente un large domaine de convergence même pour une initialisation distante de la pose désirée. La deuxième méthode est basée sur un calcul exhaustif qui permet de calculer toutes les solutions minimisant une erreur donnée. Dans cette section, nous allons appliquer les deux méthodes de calcul de pose proposées dans le cas continu au cas discret. Dans le paragraphe suivant, nous validons la méthode utilisant un AVV. Dans le paragraphe 4.4.2, nous

validons celle utilisant une table de valeur.

4.4.1 Pose par AVV

Dans le chapitre précédent, nous avons montré que le découplage réalisé dans le cas continu peut être généralisé au cas discret. Nous avons obtenu également une matrice d'interaction dans le cas discret de forme similaire à celle obtenue dans le cas continu. Le principe de calcul de pose par AVV reste donc le même dans le cas discret. Cependant, les moments après un mouvement de rotation peuvent être calculés directement par :

$$m'_{ij} = \sum_{k=0}^N \frac{(r_{11}x_k + r_{12}y_k + r_{13})^i (r_{21}x_k + r_{22}y_k + r_{23})^j}{(r_{31}x_k + r_{32}y_k + r_{33})^{i+j}} \quad (4.24)$$

L'utilisation de la méthode de calcul de pose par AVV en utilisant les moments invariants comme informations visuelles présente plusieurs intérêts par rapport aux méthodes classiques de calcul de pose utilisant les coordonnées des points. En effet :

- cette méthode exige seulement que le nombre de points dans l'image désirée et l'image courante soit le même (aucun appariement entre les points n'est exigé).
- la représentation utilisée dans la boucle de commande est minimale. En effet, le nombre d'informations visuelles utilisées pour le calcul de pose (6) est égal au nombre de degrés de liberté (6).
- le domaine de convergence obtenu est large pour des positions initiales éloignées de la position désirée. Une initialisation grossière mais appartenant à ce domaine est suffisante pour assurer la convergence.

Nous illustrons cela en utilisant les images données par la figure 4.9. Les transformations ${}^e\mathbf{M}_p$ correspondant à ces images et leurs estimées ont les valeurs suivantes :

$$\mathbf{M}_5 = \begin{pmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 0.9064 & 0.4224 & -0.2957 \\ 0 & -0.4224 & 0.9064 & -0.6345 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \widehat{\mathbf{M}}_5 = \begin{pmatrix} 1.0000 & 0.0000 & -0.0000 & 0.0000 \\ 0.0000 & 0.9064 & 0.4225 & -0.2958 \\ 0.0000 & -0.4225 & 0.9064 & -0.6345 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \\ \mathbf{U}\theta = (-25.00 \quad 0 \quad 0)^o \quad \widehat{\mathbf{U}}\theta = (-25.00 \quad 0 \quad 0)^o \quad (4.25)$$

$$\mathbf{M}_6 = \begin{pmatrix} 0.3808 & 0.8528 & 0.3575 & -0.5914 \\ -0.8528 & 0.1744 & 0.4924 & -0.4144 \\ 0.3575 & -0.4924 & 0.7936 & -0.3586 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \quad \widehat{\mathbf{M}}_6 = \begin{pmatrix} 0.4567 & 0.8569 & 0.2390 & -0.4487 \\ -0.8865 & 0.4160 & 0.2024 & 0.0607 \\ 0.0740 & -0.3043 & 0.9497 & -0.5641 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \mathbf{U}\theta = (-40.00 \quad 0 \quad -69.28)^o \quad \widehat{\mathbf{U}}\theta = (-18.27 \quad 5.95 \quad -62.87)^o \quad (4.26)$$

$$\mathbf{M}_7 = \begin{pmatrix} 0.4695 & 0 & -0.8829 & -0.7400 \\ 0 & 1.0000 & 0 & -0.0000 \\ 0.8829 & 0 & 0.4695 & -0.3200 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix} \quad \widehat{\mathbf{M}}_7 = \begin{pmatrix} 0.4695 & 0.0015 & -0.8829 & -0.7400 \\ 0.0015 & 1.0000 & 0.0005 & -0.0010 \\ 0.8829 & 0.0005 & 0.4695 & -0.3201 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$$

$$\mathbf{U}\theta = \begin{pmatrix} 0 & -62 & 0 \end{pmatrix}^o \quad \widehat{\mathbf{U}}\theta = \begin{pmatrix} 0 & -62 & 0 \end{pmatrix}^o$$
(4.27)

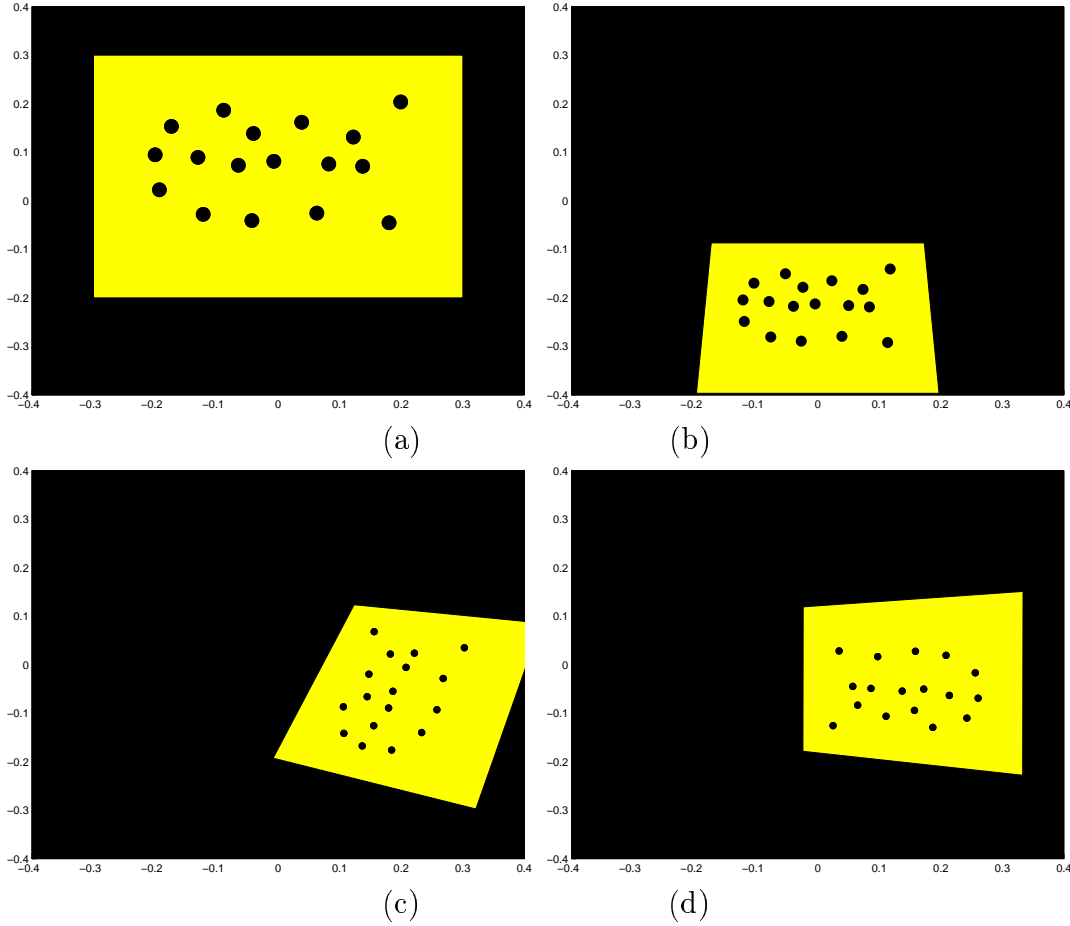


FIG. 4.9 – Images utilisées pour le calcul de pose dans le cas discret : (a) image obtenue pour la position parallèle, (b) image obtenue après la transformation \mathbf{M}_5 , (c) image obtenue après la transformation \mathbf{M}_6 , (d) image obtenue après la transformation \mathbf{M}_7

On peut noter les bons résultats obtenus pour la première et la troisième image. Par contre, le système a convergé vers un minimum local pour la seconde image (dans tous les cas la matrice de rotation a été initialisée par une rotation de $\alpha - \alpha^*$ autour de l'axe optique de la caméra). Les courbes de la figure 4.10 montrent la décroissance du module de l'erreur en utilisant les informations visuelles (r_9, r_{10}, α) pour calculer la rotation. On peut noter que la vitesse de décroissance est extrêmement lente

pour la seconde image, mais également que cette erreur converge vers une valeur extrêmement faible. Dans les autres cas où la convergence vers le minimum global a été obtenu, la vitesse de convergence est par contre bien plus rapide.

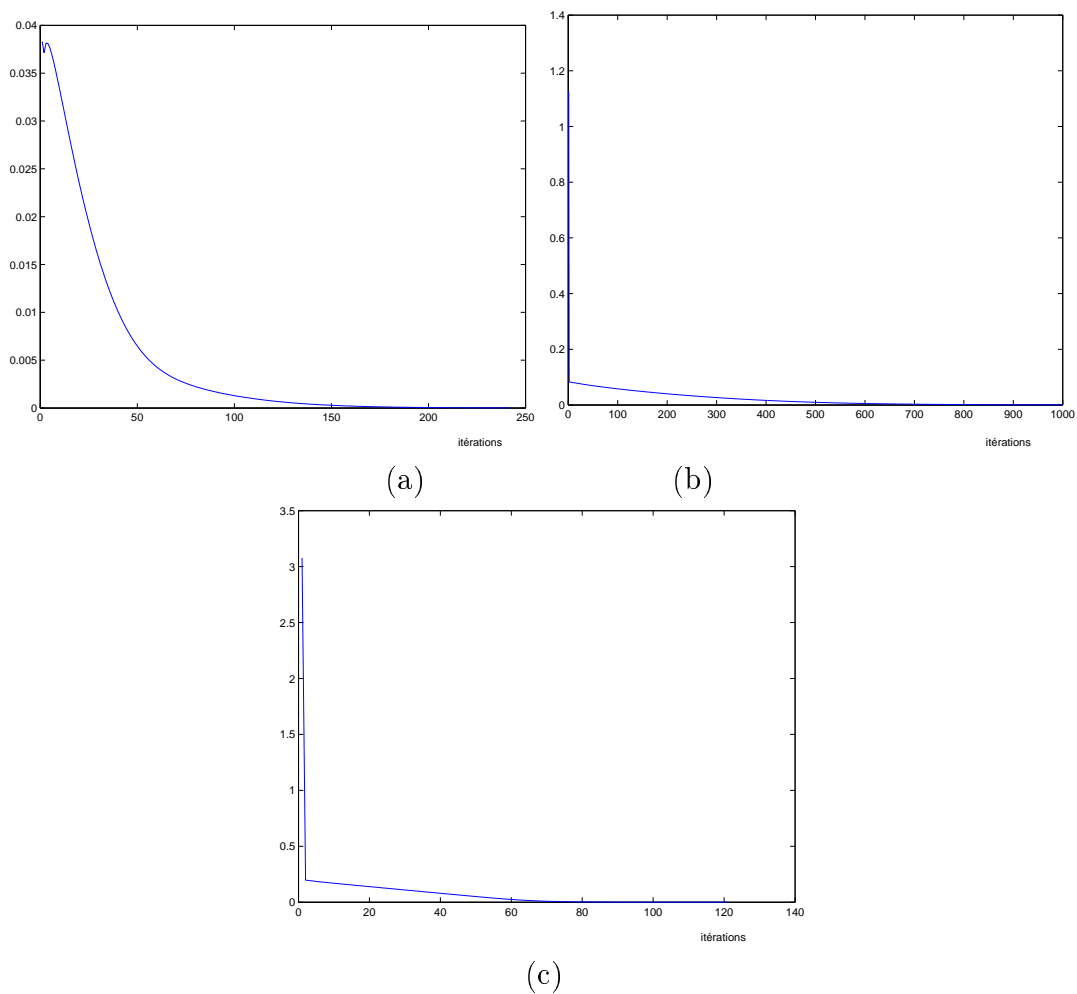


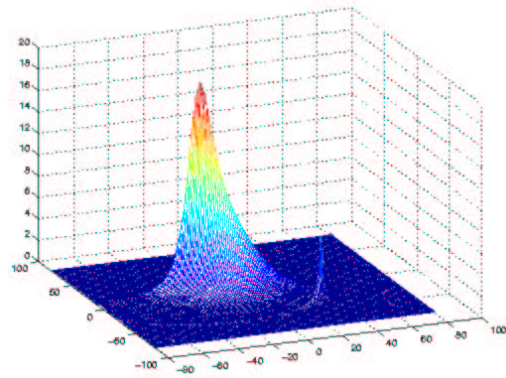
FIG. 4.10 – *Décroissance du module de l'erreur : (a) Résultat obtenu pour M_5 , (b) Résultat obtenu pour M_6 , (c) Résultat obtenu pour M_7*

4.4.2 Pose en utilisant une table de valeur

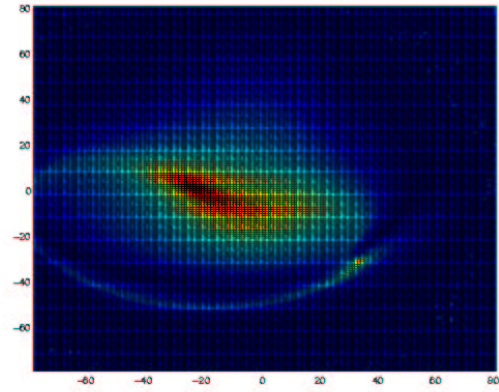
Nous traitons les mêmes cas de figures que précédemment (voir la figure 4.9). La figure 4.11 donne la variation de $f(e_r)$ en fonction des rotations appliquées sur la caméra dans sa position inconnue. Ces cartes ont été construites en considérant des rotations avec un pas égal à 1 degré autour d'un axe appartenant au plan formé par les deux axes de la caméra X et Y et dans l'intervalle $[-80^\circ, 80^\circ]$. Les valeurs de la pose obtenues en utilisant ces tables sont identiques aux vraies valeurs.

Les cartes de la fonction $f(e_r)$ montrent l'existence d'un minimum global et d'au plus deux minima locaux. L'un des minima locaux est presque symétrique au minimum global par rapport au centre de la carte. Le temps de calcul de chaque carte pour l'ensemble de points considéré est de l'ordre de 0.4 seconde sur un processeur de 2 GHz. Ce temps de calcul peut être largement diminué sachant que les éléments de cette table d'erreur peuvent être calculés en parallèle en utilisant un système multiprocesseur ou si le domaine de recherche est moins large. Cette méthode peut fournir également une bonne initialisation à la méthode du calcul de pose par AVV.

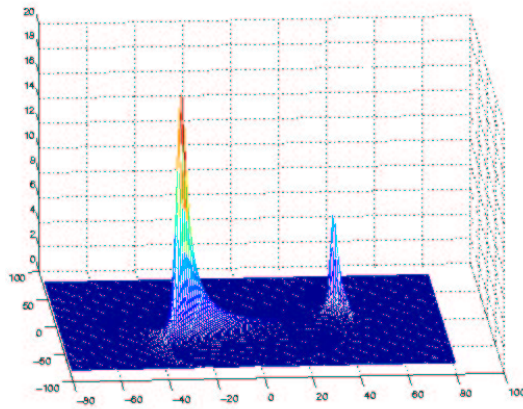
Comme nous l'avons vu, les deux méthodes de calcul de pose proposées dans ce chapitre estiment dans une première étape le mouvement de rotation. En pratique, cette étape ne nécessite pas la connaissance du modèle de l'objet. En effet, l'estimation du mouvement de rotation de la pose suppose juste la connaissance des invariants r_i et θ pour une position de la caméra où les plans image et objet sont parallèles. Si l'on ne dispose pas du modèle de l'objet, le mouvement de translation peut être déterminé également, mais à un facteur d'échelle près. Plus précisément, le facteur d'échelle est donné par la distance Z^* qui intervient dans les informations visuelles proposées pour contrôler les mouvements de translation. Dans la section suivante, nous proposons une nouvelle méthode basée également sur les moments invariants pour déterminer les valeurs des moments pour une image de l'objet dans une position parallèle au plan image sans disposer d'un modèle de l'objet.



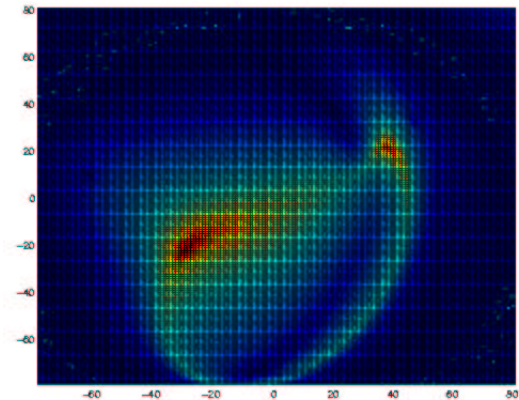
(a)



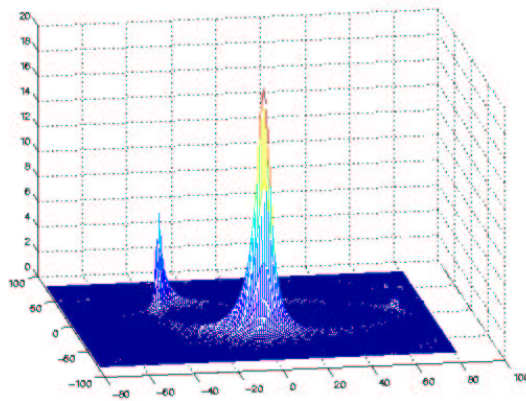
(b)



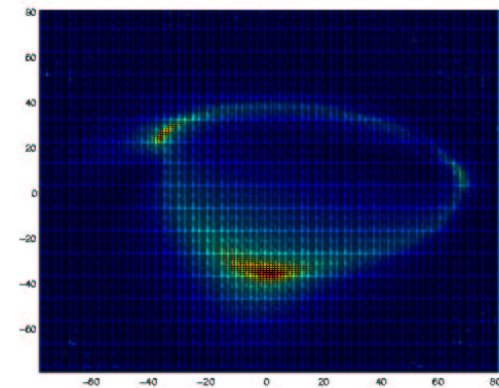
(c)



(d)



(e)



(f)

FIG. 4.11 – Représentation de $f(e_r)$ en amplitude et avec des niveaux de couleur : (a),(b) Résultat obtenu pour \mathbf{M}_5 , (c),(d) Résultat obtenu pour \mathbf{M}_6 , (e),(f) Résultat obtenu pour \mathbf{M}_7

4.5 Calcul partiel du déplacement 3D sans modèle de l'objet

Dans la suite, nous présentons une nouvelle méthode de calcul de pose partielle en utilisant deux images de l'objet acquises en deux positions différentes. Dans un premier temps, nous donnons le principe généralement utilisé pour les méthodes existantes qui permettent de résoudre ce problème.

4.5.1 État de l'art du calcul de pose sans modèle de l'objet

L'estimation du déplacement d'une caméra entre deux prises de vue (voir figure 4.12), lorsqu'on ne dispose pas d'informations précises sur la géométrie de la scène, est un problème classique en vision par ordinateur. Cependant, la majorité des approches proposées utilisent des primitives géométriques simples de type coordonnées de point ou segments de droite [Malis 99, Kanatani 99]. Ces approches se basent généralement sur le calcul d'une matrice d'homographie, définie par:

$$\mathbf{Z}'\mathbf{m}' = \mathbf{Z}\mathbf{H}\mathbf{m} \quad (4.28)$$

avec :

$$\mathbf{H} = \mathbf{R} + \mathbf{t}\mathbf{n}^T/d = \mathbf{R} + \mathbf{t} \begin{pmatrix} A & B & C \end{pmatrix}$$

où A , B et C sont les paramètres du plan objet dans le repère \mathcal{F}_{c_1} , \mathbf{n} est le vecteur normal au plan objet défini également dans \mathcal{F}_{c_1} , $\mathbf{m} = (x, y, 1)$ et $\mathbf{m}' = (x', y', 1)$ les coordonnées homogènes du point image définis respectivement dans les repères \mathcal{F}_{c_1} et \mathcal{F}_{c_2} . A partir de la matrice \mathbf{H} , on peut déterminer la matrice de rotation \mathbf{R} , le vecteur de translation à un facteur d'échelle, et le vecteur \mathbf{n} si $\mathbf{t} \neq \mathbf{0}$. Dans le paragraphe suivant, nous présentons une nouvelle méthode de calcul des différents paramètres mentionnés précédemment sans avoir recours au calcul de la matrice d'homographie.

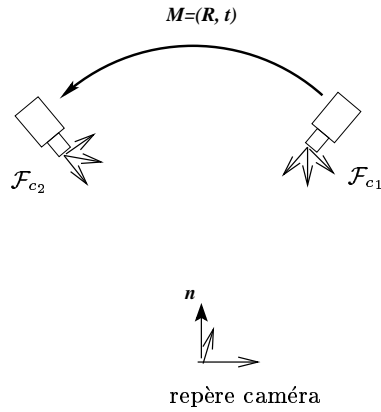


FIG. 4.12 – Calcul de pose partielle à partir de deux images

4.5.2 Méthode proposée

Le principe de la méthode que nous proposons est similaire à celui utilisé par la méthode de calcul de pose en utilisant une table de valeur. En effet, pour déterminer la rotation entre les deux positions de la caméra, ainsi que l'orientation de l'objet dans un des repères caméra, nous cherchons les deux mouvements de rotation à appliquer sur les deux positions de la caméra qui minimisent l'erreur $e_r = \sum_i (r_i - r_i^*)^2$ où r_i et r_i^* sont calculés respectivement à partir de l'image 1 et l'image 2 et les mouvements de rotation appliquées sur chaque repère caméra. Les valeurs obtenues pour ces rotations permettent de ramener les deux caméras à des positions telles que les plans objet et image soient parallèles. Un résumé de notre algorithme est donné par le schéma 4.13. Dans le paragraphe suivant, nous validons cette approche dans le cas continu, ainsi que dans le cas discret.

calcul des invariants r_i pour un élément (k, l) et pour les deux positions de la caméra

- vecteur de rotation :

$$\mathbf{u} = (1 \ 0 \ 0) \text{ si } k = l = 0$$

$$\mathbf{u} = \left(\frac{k}{\sqrt{k^2+l^2}} \quad \frac{l}{\sqrt{k^2+l^2}} \quad 0 \right) \text{ si non}$$

- angle de rotation $\theta = \sqrt{k^2 + l^2} \Delta\theta$:

- matrice de rotation $\mathbf{R} = \mathbf{I}_3 + \sin \theta [\mathbf{u}]_{\times} + (1 - \cos \theta) [\mathbf{u}]_{\times}^2$
avec

$$[\mathbf{u}]_{\times} = \begin{pmatrix} 0 & -u(3) & u(2) \\ u(3) & 0 & -u(1) \\ -u(2) & u(1) & 0 \end{pmatrix}$$

et

$$[\mathbf{u}]_{\times}^2 = \begin{pmatrix} u(1)^2 - 1 & u(1)u(2) & u(1)u(3) \\ u(1)u(2) & u(2)^2 - 1 & u(2)u(3) \\ u(1)u(3) & u(2)u(3) & u(3)^2 - 1 \end{pmatrix}$$

- calcul des valeurs des moments après la rotation \mathbf{R} pour chaque position de la caméra.
- calcul des informations visuelles $r_i(k, l)$ après la rotation \mathbf{R} à partir des moments pour les deux positions de la caméra.

Détermination des mouvements de rotation

- recherche systématique des couples (k_1, l_1) et (k_2, l_2) qui minimisent l'erreur

$$e_r = \sum_i (r_{1_i}(k_1, l_1) - r_{2_i}(k_2, l_2))^2.$$

où $r_{1_i}(k_1, l_1)$ et $r_{2_i}(k_2, l_2)$ sont les valeurs des invariants r_i calculées respectivement pour les deux positions de la caméra et pour les éléments (k_1, l_1) et (k_2, l_2) .

- calculer la normale au plan objet définie dans le repère \mathcal{F}_{c_1} par :

$$\mathbf{n} = \left(\frac{k_1}{\sqrt{k_1^2+l_1^2}} \quad \frac{l_1}{\sqrt{k_1^2+l_1^2}} \quad 0 \right)$$

- calcul de la translation après avoir amené les deux caméras à des positions parallèles par rapport à l'objet par :

$$\begin{cases} t_x &= x_n - x_n^* \\ t_y &= y_n - y_n^* \\ t_z &= a_n - a_n^* \end{cases}$$

FIG. 4.13 – Calcul de la pose partielle

4.5.3 Résultats expérimentaux

Nous traitons dans un premier temps le cas d'un objet continu. Puis, nous validons notre approche au cas discret.

4.5.3.1 Cas continu

Les images utilisées sont données sur la figure 4.14. Les valeurs des mouvements de rotation estimées permettant de se ramener à une position où les plans objet et image sont parallèles sont données par les équations (4.29), (4.30) et (4.31). Le premier résultat a été obtenu en utilisant les deux images données par les figures 4.14.a et 4.14.b, le deuxième résultat a été obtenu en utilisant les deux images données par les figures 4.14.a et 4.14.c, alors que le troisième résultat a été obtenu pour les deux images données par les figures 4.14.b et 4.14.c. On peut noter à partir de ces équations que les valeurs estimées coïncident avec les valeurs réelles.

$$\begin{aligned}\widehat{\mathbf{U}}_{\theta_1} &= [0 \ 0 \ 0] \\ \widehat{\mathbf{U}}_{\theta_2} &= [-30 \ 0 \ 0]\end{aligned}\quad (4.29)$$

$$\begin{aligned}\widehat{\mathbf{U}}'_{\theta_1} &= [0 \ 0 \ 0] \\ \widehat{\mathbf{U}}'_{\theta_2} &= [0 \ -30 \ 0]\end{aligned}\quad (4.30)$$

$$\begin{aligned}\widehat{\mathbf{U}}''_{\theta_1} &= [-30 \ 0 \ 0] \\ \widehat{\mathbf{U}}''_{\theta_2} &= [0 \ -30 \ 0]\end{aligned}\quad (4.31)$$

L'équation (4.32) donne la valeur du vecteur :

$$\mathbf{r} = [r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6 \ r_7 \ r_8 \ r_9 \ r_{10}]$$

regroupant les invariants donnés par l'équation (3.101) et calculés pour l'image 4.14.a (cette image correspond à une position de l'objet parallèle au plan image). Les valeurs de ce vecteur calculées à partir des images 4.14.c et 4.14.d obtenues après application des rotations estimées par notre approche pour se ramener à une position parallèle sont données par les équations (4.33) et (4.34). On peut noter à partir de ces équations que les valeurs r_i sont quasiment les mêmes que celles obtenues pour une position parallèle. Cela valide notre démarche. Dans le paragraphe suivant nous validons cette approche pour un objet discret.

$$r = [1.855 \ 3.201 \ 0.215 \ -0.427 \ 1.256 \ 3.309 \ 41.226 \ -4.513 \ 1.447 \ 1.729] \quad (4.32)$$

$$r' = \begin{bmatrix} 1.851 & 3.195 & 0.215 & -0.423 & 1.260 & 3.332 & 41.231 & -4.512 & 1.449 & 1.730 \end{bmatrix} \quad (4.33)$$

$$r'' = \begin{bmatrix} 1.847 & 3.196 & 0.215 & -0.425 & 1.260 & 3.329 & 41.223 & -4.510 & 1.441 & 1.723 \end{bmatrix} \quad (4.34)$$

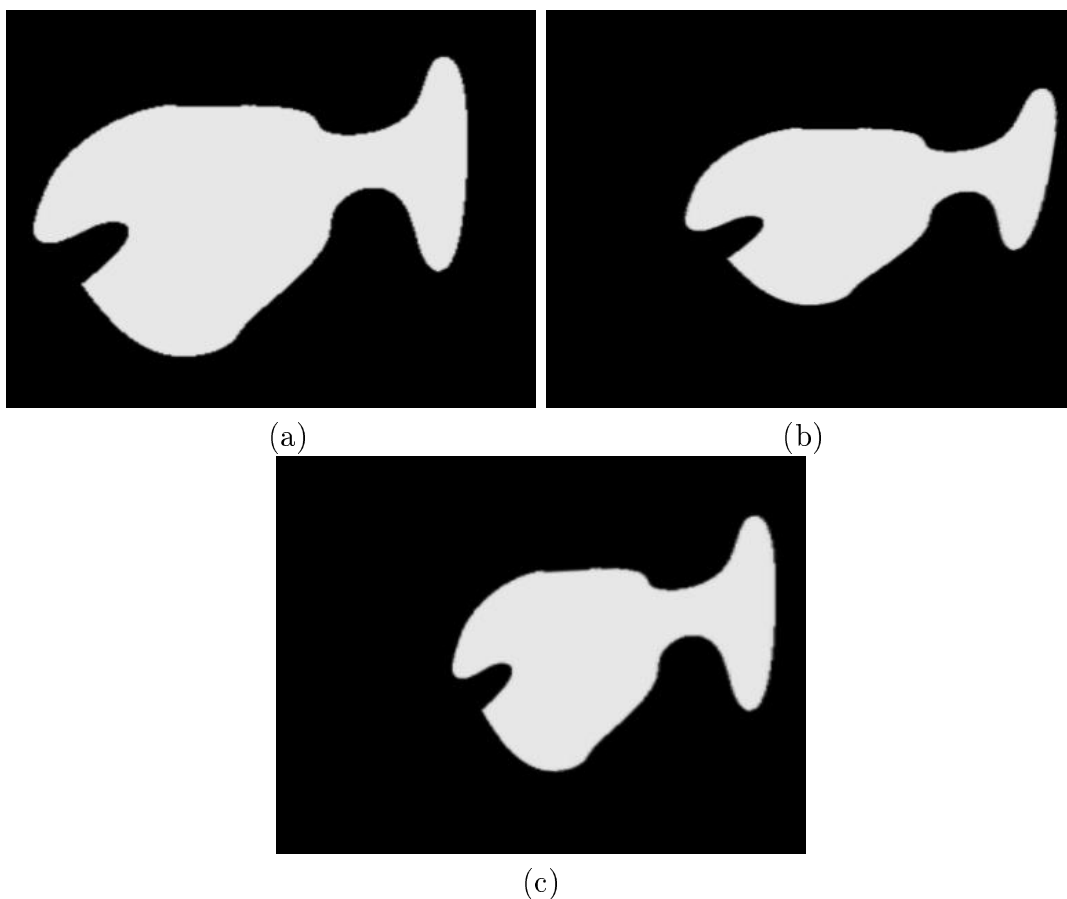


FIG. 4.14 – Images utilisées pour le calcul de pose partielle sans modèle de l'objet : (a) image obtenue pour une position parallèle (b) image obtenue après un mouvement de rotation de 30° autour de l'axe x , (c) image obtenue après un mouvement de rotation de 30° autour de l'axe y

4.5.3.2 Cas discret

Dans ce paragraphe, nous validons notre méthode par l'exemple donné sur la figure 4.15. La valeur réelle de la matrice de rotation entre les deux positions de la caméra est donnée par l'équation (4.35). Celle estimée en utilisant notre approche

est donnée par l'équation (4.36). On peut noter que la valeur estimée et la valeur réelle sont quasiment les mêmes. Par ailleurs, nous présentons dans l'équation (4.37) le résultat obtenu à partir de l'estimation de l'homographie reliant les deux images. Cette estimation a été obtenue par simple résolution d'un système linéaire, puis une décomposition de la matrice d'homographie en introduisant la contrainte d'orthonormalité de la matrice de rotation. On peut noter que le résultat obtenu par notre méthode est plus précis. Cela est dû au fait que, dans notre méthode, la contrainte d'orthonormalité de \mathbf{R} est introduite directement dans la minimisation de l'erreur entre les informations visuelles.

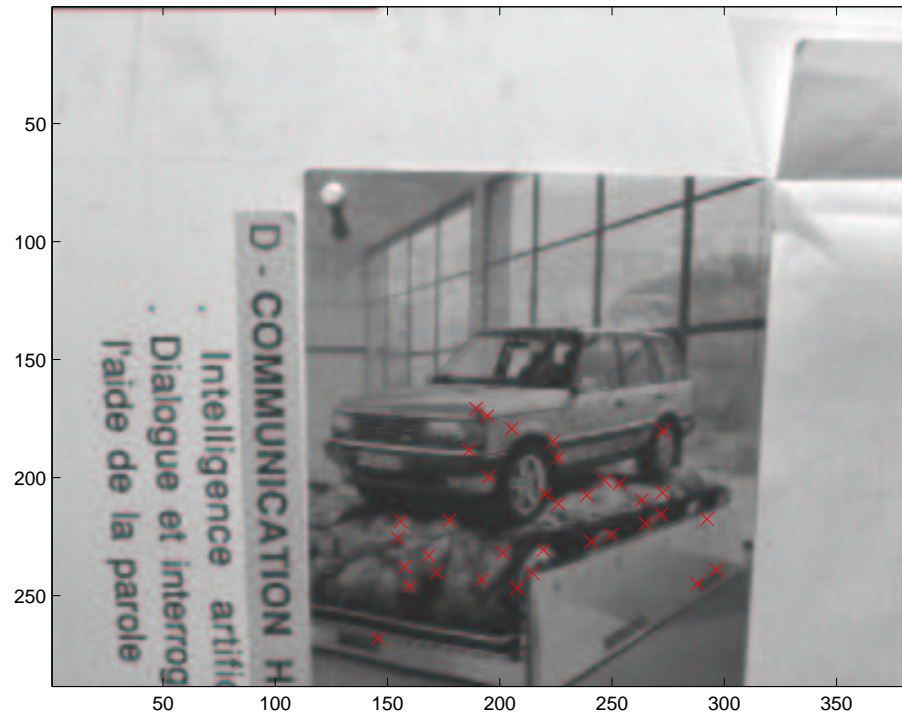
$$\begin{aligned} {}^{c_1}\mathbf{R}_{c_2} &= \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 0.9848 & 0.1736 \\ 0 & -0.1736 & 0.9848 \end{bmatrix} \\ \widehat{\mathbf{U}}\theta &= [-10 \quad 0 \quad 0] \end{aligned} \quad (4.35)$$

$$\begin{aligned} {}^{c_1}\widehat{\mathbf{R}}_{c_2} &= \begin{bmatrix} 0.9999 & 0.0003 & -0.0043 \\ 0.0003 & 0.9848 & 0.1735 \\ 0.0043 & -0.1735 & 0.9848 \end{bmatrix} \\ \widehat{\mathbf{U}}\theta &= [-9.99 \quad -0.24 \quad 0] \end{aligned} \quad (4.36)$$

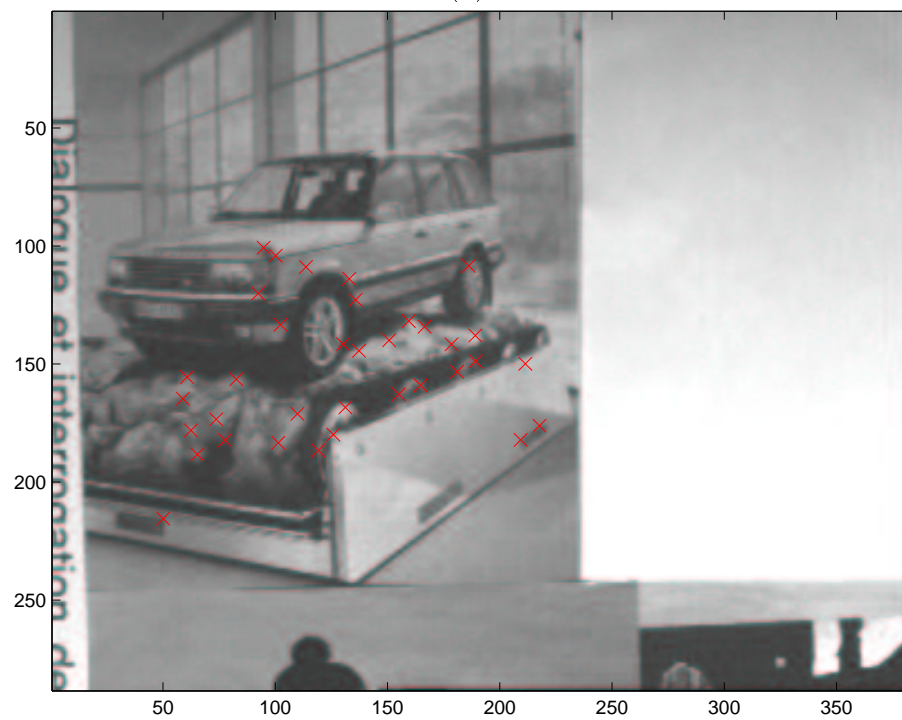
$$\begin{aligned} {}^{c_1}\widehat{\mathbf{R}}'_{c_2} &= \begin{bmatrix} 0.9986 & 0.0389 & 0.0355 \\ -0.0454 & 0.9769 & 0.2084 \\ -0.0266 & -0.2097 & 0.9773 \end{bmatrix} \\ \widehat{\mathbf{U}}\theta &= [-12.07 \quad 1.79 \quad -2.43] \end{aligned} \quad (4.37)$$

4.6 Conclusion

Dans ce chapitre, nous avons proposé deux méthodes du calcul de pose basées sur les moments invariants. La première méthode utilise un asservissement visuel virtuel pour estimer le déplacement entre une position connue de la caméra et une autre position à déterminer. La deuxième méthode que nous avons proposée se base sur un calcul exhaustif d'une table d'erreurs pour le calcul de pose. Cela rend cette dernière plus lente que la première méthode. Pour accélérer le calcul des deux méthodes, on a proposé une méthode basée sur un développement de Taylor. Toutefois, le temps de calcul de pose en utilisant la deuxième méthode peut encore être réduit étant donné que le calcul des éléments de la table d'erreur peut se faire en parallèle (à l'aide d'un système multiprocesseurs). En asservissement visuel, le calcul de la commande se fait en temps réel. L'utilisation de la première méthode de calcul de pose pendant l'asservissement visuel est alors plus judicieuse, d'autant plus que le résultat obtenu sur une image peut servir d'initialisation pour estimer la pose correspondant à l'image suivante. Néanmoins, la méthode se basant sur un calcul exhaustif de la pose peut être utilisée avant de lancer l'asservissement visuel pour calculer la pose de la caméra qui correspond à la première image acquise. En effet, cette méthode permet d'avoir une carte de toutes les solutions pour la pose minimisant l'erreur entre deux positions de la caméra. Cela permettra d'obtenir une bonne initialisation pour le minimum global, ainsi qu'une carte des minima locaux. Cela peut être utilisé pour éviter ces minima locaux en asservissement visuel, ainsi que pour planifier une trajectoire dans le domaine 3D de la caméra. On peut également s'en servir pour choisir les deux informations visuelles r_i et r_j servant à contrôler les vitesses de rotations ω_x et ω_y . Rajoutons que la précision de la méthode utilisant la table d'erreurs est liée au pas entre deux rotations consécutives, et une diminution de ce pas augmente le temps de calcul. Toutefois, dans le cas d'une cible de points, le temps de calcul de la carte d'erreur reste très raisonnable. En utilisant un système multiprocesseurs cette méthode peut être utilisée dans des systèmes temps réel qui n'exigent pas une estimation avec une haute précision de la pose. Dans le cas contraire, l'utilisation de la deuxième méthode pour initialiser la première s'avère judicieuse. Les deux méthodes de calcul de pose proposées dans ce chapitre estiment dans une première étape le mouvement de rotation. En pratique, cette étape ne nécessite pas la connaissance du modèle de l'objet. Si l'on ne dispose pas du modèle de l'objet, le mouvement de translation peut être déterminé également, mais à un facteur d'échelle près. Plus précisément, le facteur d'échelle est donné par la distance Z^* qui intervient dans la variation des informations visuelles proposées pour contrôler les mouvements de translation. Nous en avons déduit une nouvelle méthode de calcul partiel du déplacement 3D. Notre méthode s'avère plus précise que la méthode se basant sur un calcul d'homographie.



(a)



(b)

FIG. 4.15 – Images utilisées pour le calcul de pose partielle sans modèle de l'objet dans le cas discret: (a) image 1 (b) image 2

Chapitre 5

Validation expérimentale en asservissement visuel

Dans ce chapitre, nous validons les résultats théoriques sur le calcul des invariants et le choix des informations visuelles par des expériences réalisées en utilisant un système robotique à six degrés de liberté. Dans les séries d'expériences que nous présentons dans ce chapitre, nous avons principalement utilisé des objets continus et des objets discrets de forme asymétriques. Néanmoins, nous présentons également quelques résultats obtenus pour une objet symétrique. Dans le cas d'un objet non symétrique, nous utiliserons en général les six combinaisons de moments proposés dans le chapitre 3 pour contrôler les six degrés de liberté de la caméra :

$$\mathbf{s} = (x_n, y_n, a_n, r_i, r_j, \alpha) \quad (5.1)$$

avec :

$$\begin{cases} x_n = a_n x_g, y_n = a_n y_g, a_n = Z^* \sqrt{\frac{a^*}{a}} \\ \alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \end{cases}$$

où $a = m_{00}$ dans le cas continu et $a = \mu_{20} + \mu_{02}$ dans le cas discret. Le choix des invariants r_i et r_j n'as pas encore été fixé. C'est pourquoi nous discutons du choix de ces invariants selon l'objet considéré en utilisant les résultats obtenus dans le chapitre du calcul de pose en utilisant un algorithme d'optimisation exhaustif.

5.1 Choix des invariants r_i et r_j

Dans ce paragraphe, nous discutons du choix des couples d'invariants (r_i, r_j) à introduire dans la boucle de commande selon l'objet considéré. Nous choisissons les deux invariants qui permettent d'avoir à la fois un bon conditionnement de la matrice d'interaction et un bon comportement de l'erreur $e_r = (r_i - r_i^*)^2 + (r_j - r_j^*)^2$ en fonction des rotations ω_x et ω_y autour de la position désirée. En effet, nous cherchons une combinaison d'invariants telle que l'erreur $e_r = (r_i - r_i^*)^2 + (r_j - r_j^*)^2$ en fonction des rotations ω_x et ω_y présente un minimum global tel que la zone d'influence associée à ce minimum soit la plus large et la plus symétrique possible. Le calcul de l'erreur e_r en fonction des rotations autour des axes x et y de la caméra se fait de la même manière que dans la méthode de calcul de pose utilisant un calcul exhaustif. Les notions de minima et zone d'influence sont les mêmes que celles utilisées en topographie pour calculer la ligne de partage des eaux et en morphologie mathématique pour segmenter une image. Un bref rappel de ces notions est donné en annexe B dans le cas d'une image. Plus de détails sont donnés dans [Vincent 92]. En fait, obtenir une large zone d'influence associée au minimum global assurera un large domaine de convergence du système autour de la position désirée. Dans la suite, nous essayons d'abord de choisir parmi les couples r_i, r_j ceux qui donnent un bon comportement de e_r . Dans le cas où aucun couple n'amène à un résultat satisfaisant, nous choisissons des combinaisons de plus de deux invariants présentant un bon comportement de $e_r = \sum_i (r_i - r_i^*)^2$. Comme cela est bien connu [Chaumette 98], un nombre redondant d'informations visuelles peut potentiellement amener le système vers un minimum local, mais, comme nous avons vu qu'en utilisant un nombre minimal et donc non redondant les minima locaux pouvaient exister également, il est préférable de choisir un nombre d'invariants tel que la zone d'influence au minimum global soit la plus grande possible. Toutefois, dans les expérimentations que nous avons effectuées, et qui sont décrites à la fin de ce chapitre, nous n'avons jamais eu besoin de recourir à un tel choix d'invariants redondants.

Dans la suite de ce chapitre, nous présentons des résultats sur le choix pour les trois objets donnés par les figures 5.1.a, 5.1.b et 5.1.c. Nous présentons également des résultats expérimentaux pour l'image donnée par la figure 5.1.d

Dans le cas de l'objet 5.1.a ("la baleine"), plusieurs combinaisons d'invariants donnent un bon comportement de l'erreur. En effet, comme on peut le voir à partir des figures 5.2 à 5.5, parmi les bonnes combinaisons, on trouve les ensembles (r_9, r_{10}) , (r_9, r_5) , (r_{10}, r_4) , (r_{10}, r_5) qui correspondent aux figures 5.2.a, 5.4.b, 5.4.c et 5.4.d. On peut signaler que certaines combinaisons sont à proscrire, telles par exemple (r_6, r_3) et (r_3, r_4) (voir figures 5.2.c et 5.3.b). Dans le cas de l'objet donné par la figure 5.1.b, les ensembles d'invariants $(r_6, r_3, r_2, r_4, r_5)$, (r_6, r_4) , (r_3, r_5) , (r_2, r_4) , $(r_9, r_{10}, r_5, r_4, r_2)$ et $(r_9, r_{10}, r_7, r_8, r_6, r_3, r_2, r_4, r_5)$ présentent un bon comportement de e_r (voir figures 5.6.b, 5.6.e, 5.7.c, 5.7.d, 5.9.e et 5.9.f). Les résultats obtenus dans le cas de l'objet 5.1.c. sont donnés sur les figures 5.10 à 5.13. À partir de ces figures, on note que plusieurs combinaisons d'invariants présentent un bon comportement de e_r ((r_9, r_{10}) ,

$(r_6, r_3, r_2, r_4, r_5)$, (r_6, r_4) , (r_2, r_4) , (r_{10}, r_2)) qui correspondent aux figures 5.10.a, 5.10.b, 5.11.d, 5.13.a. Dans la suite, nous utiliserons ces combinaisons d'invariants en asser-vissement visuel.

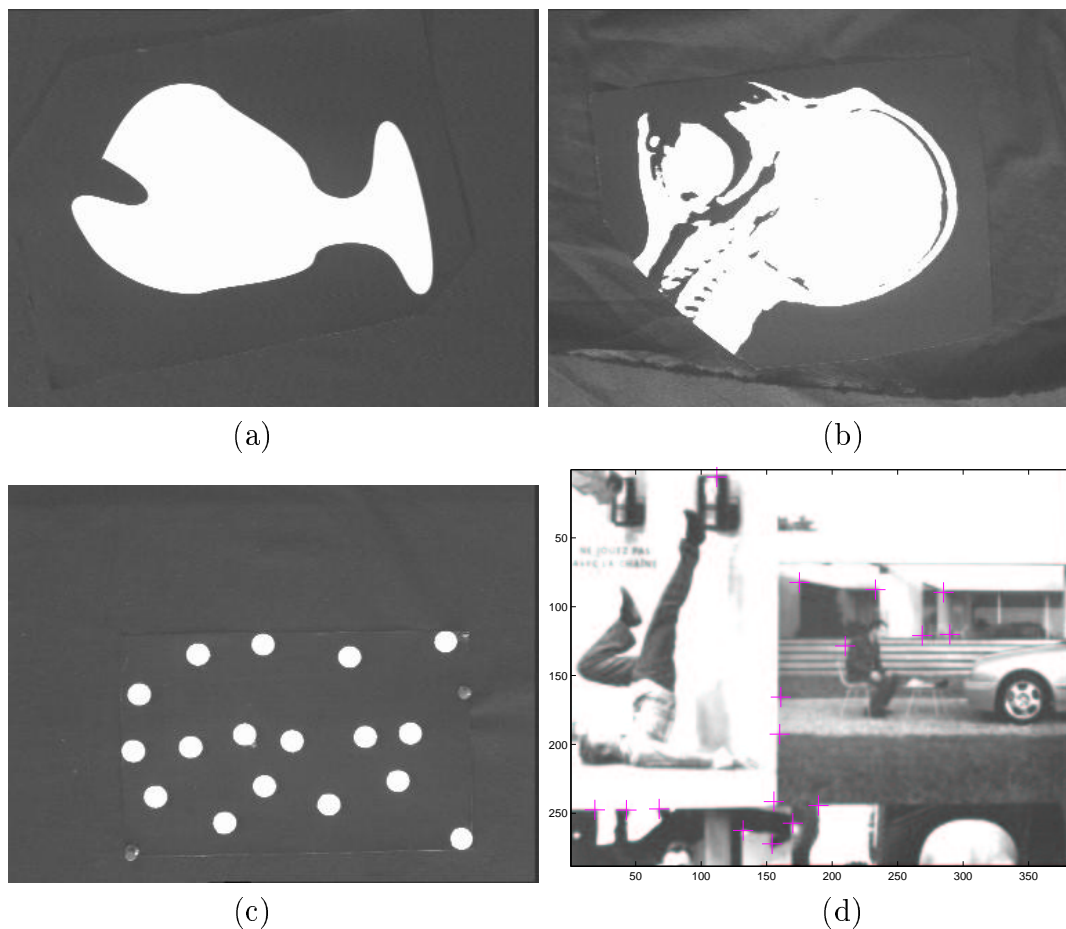
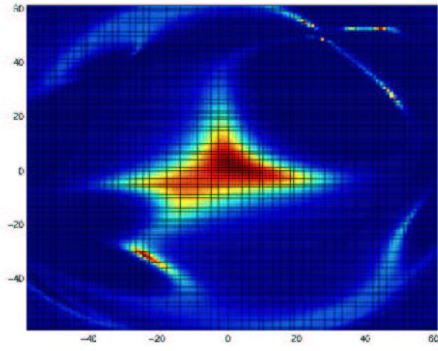
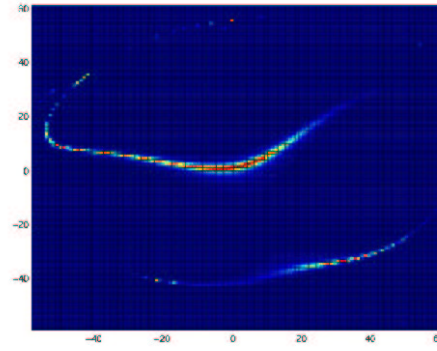


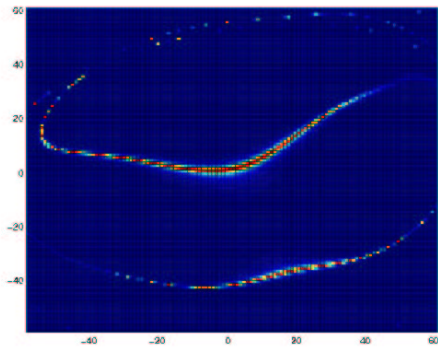
FIG. 5.1 – Image d'objets utilisés dans les expérimentations : (a) image d'une "ba-leine", (b) image d'un "cerveau", (c) ensemble de points, (d) ensemble de points extrait en utilisant un détecteur de Harris.



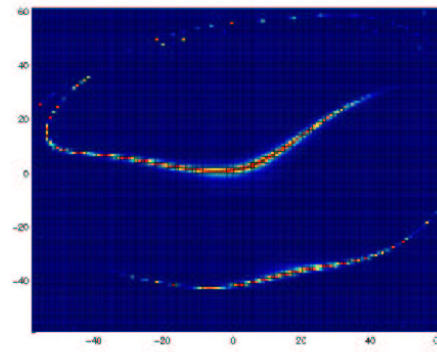
(a)



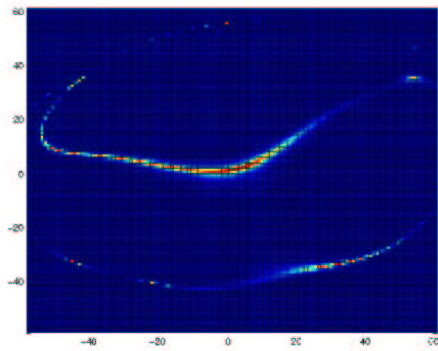
(b)



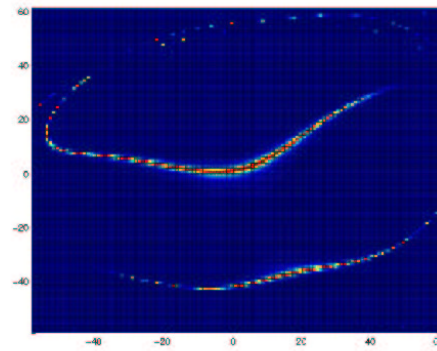
(c)



(d)



(e)



(f)

FIG. 5.2 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour la “baleine”: (a) (r_9, r_{10}) , (b) $(r_6, r_3, r_2, r_4, r_5)$, (c) (r_6, r_3) , (d) (r_6, r_2) , (e) (r_6, r_4) , (f) (r_6, r_5)

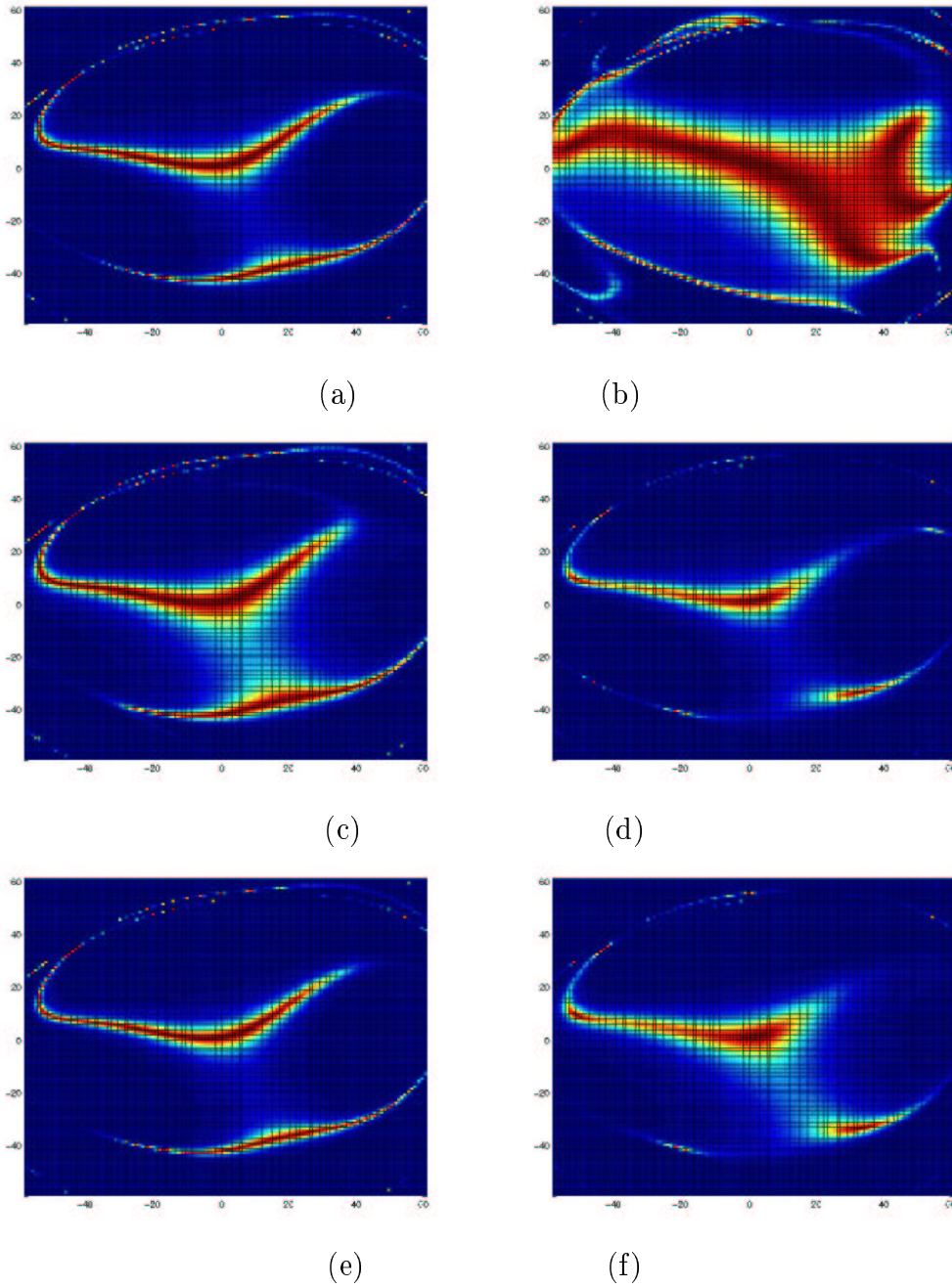
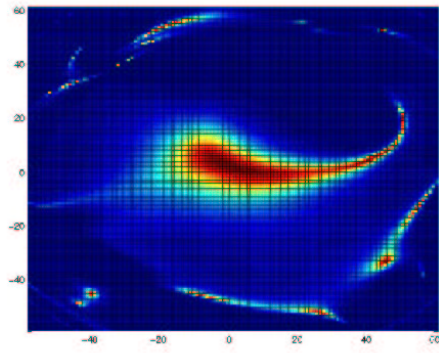
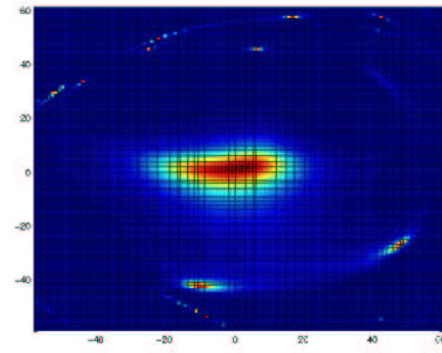


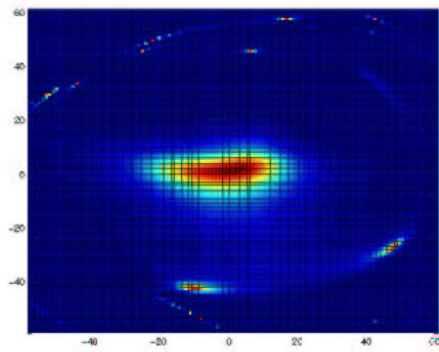
FIG. 5.3 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour la “baleine” : (a) (r_3, r_2) , (b) (r_3, r_4) , (c) (r_3, r_5) , (d) (r_2, r_4) , (e) (r_2, r_5) , (f) (r_4, r_5)



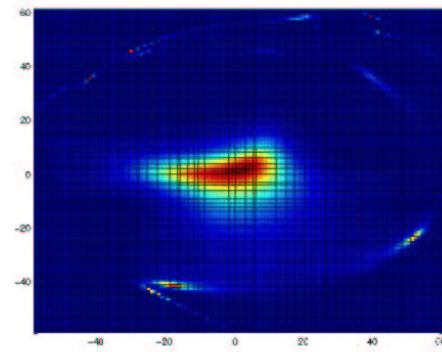
(a)



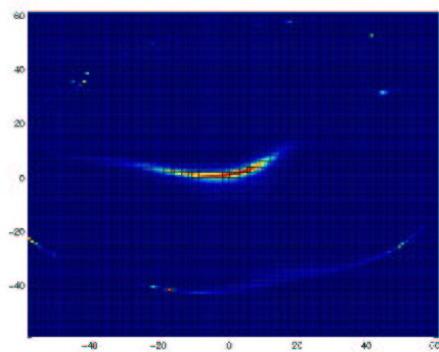
(b)



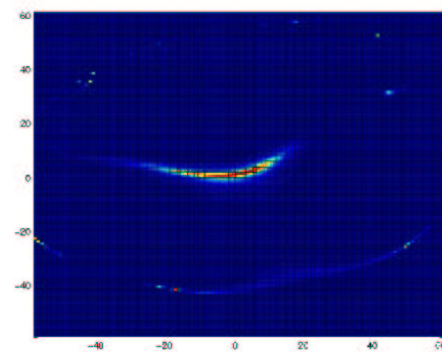
(c)



(d)



(e)



(f)

FIG. 5.4 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour la “baleine” : (a) (r_9, r_4) , (b) (r_9, r_5) , (c) (r_{10}, r_4) , (d) (r_{10}, r_5) , (e) (r_{10}, r_6) , (f) (r_{10}, r_3)

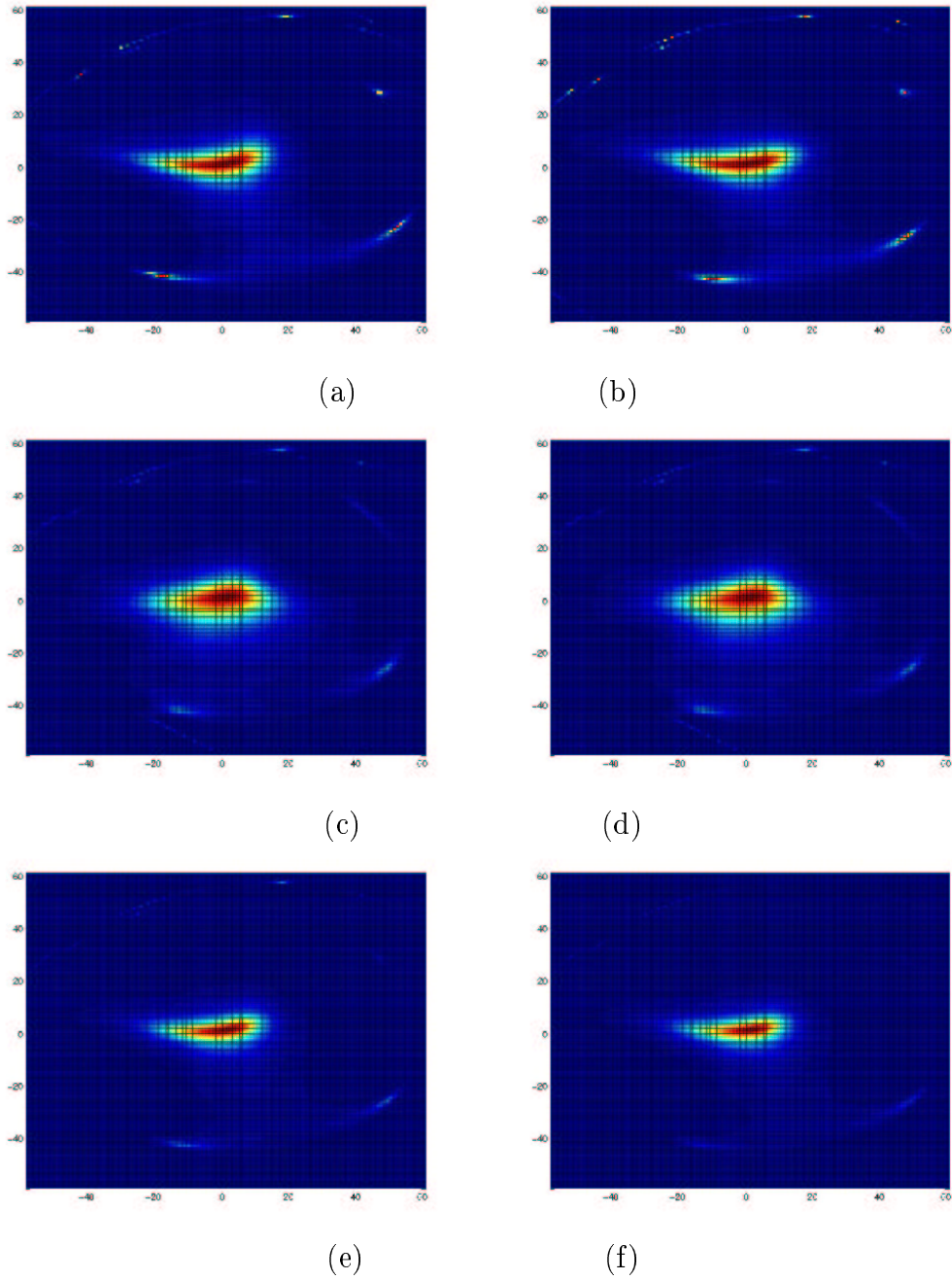


FIG. 5.5 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour la “baleine”: (a) (r_{10}, r_2) , (b) (r_9, r_2) , (c) (r_9, r_{10}, r_5) , (d) (r_9, r_{10}, r_5, r_2) , (e) $(r_9, r_{10}, r_5, r_4, r_2)$, (f) $(r_9, r_{10}, r_7, r_8, r_6, r_3, r_2, r_4, r_5)$

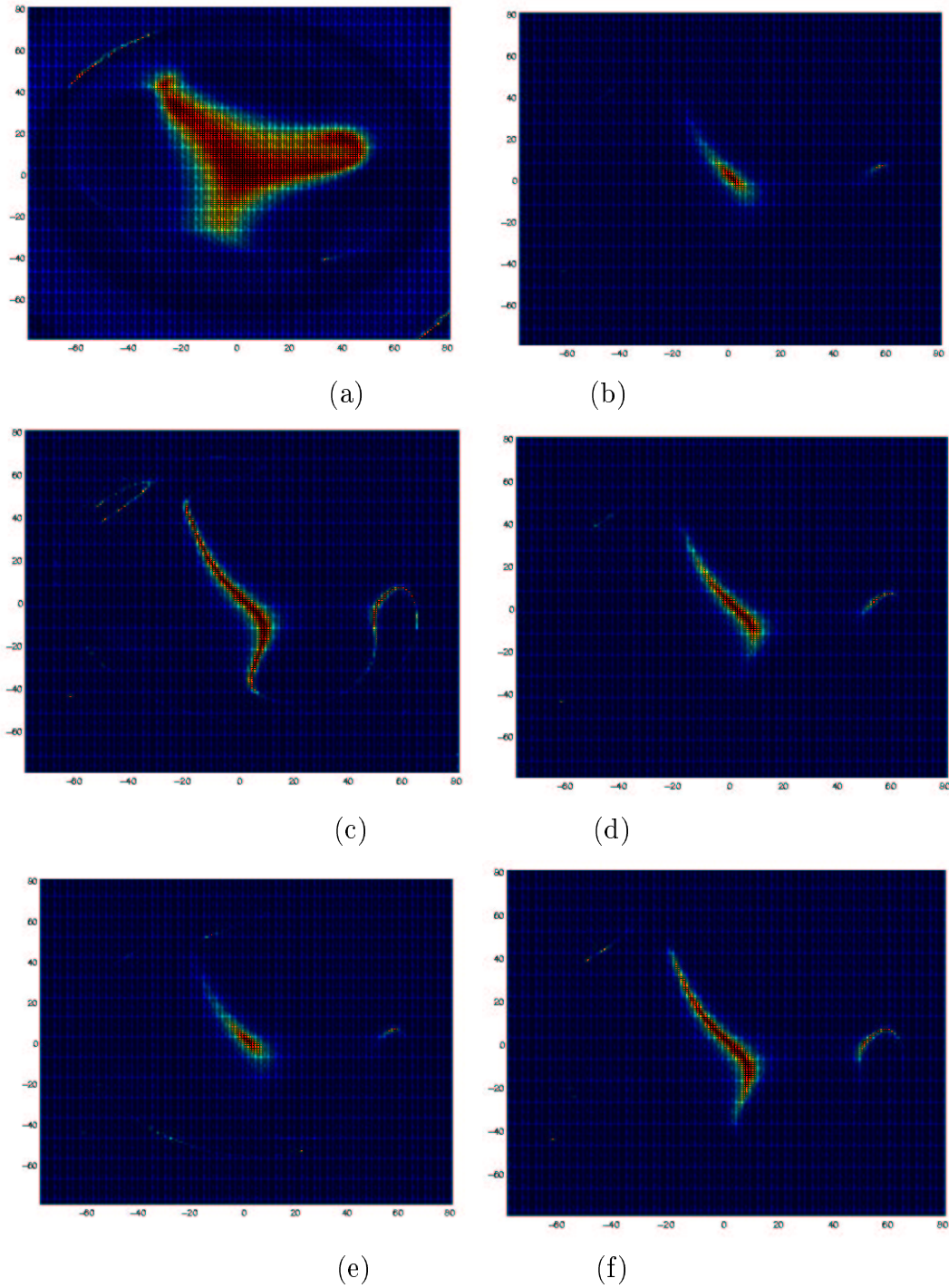


FIG. 5.6 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour le “cerveau” : (a) (r_9, r_{10}) , (b) $(r_6, r_3, r_2, r_4, r_5)$, (c) (r_6, r_3) , (d) (r_6, r_2) , (e) (r_6, r_4) , (f) (r_6, r_5)

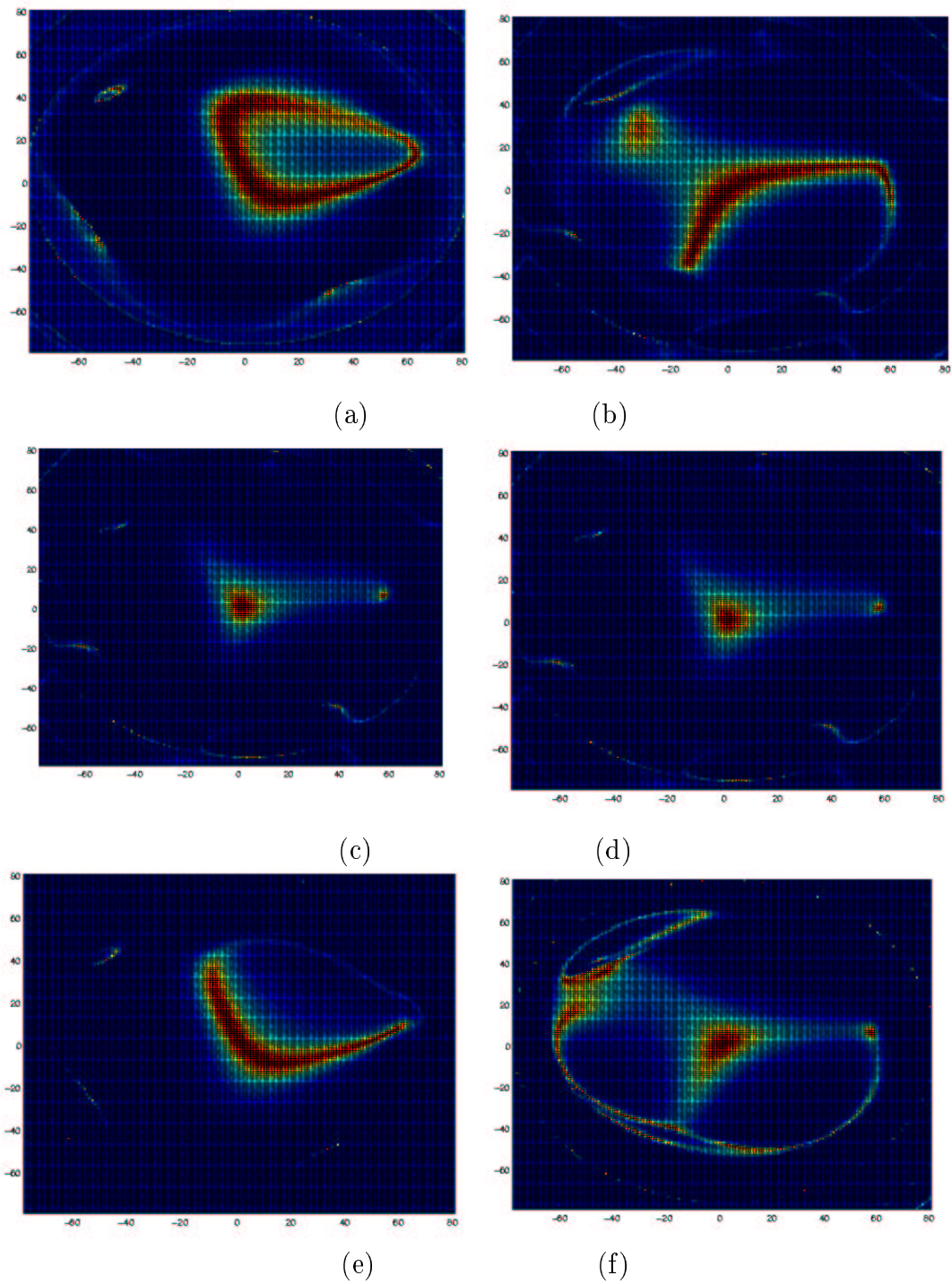


FIG. 5.7 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour le “cerveau” : (a) (r_3, r_2) , (b) (r_3, r_4) , (c) (r_3, r_5) , (d) (r_2, r_4) , (e) (r_2, r_5) , (f) (r_4, r_5)

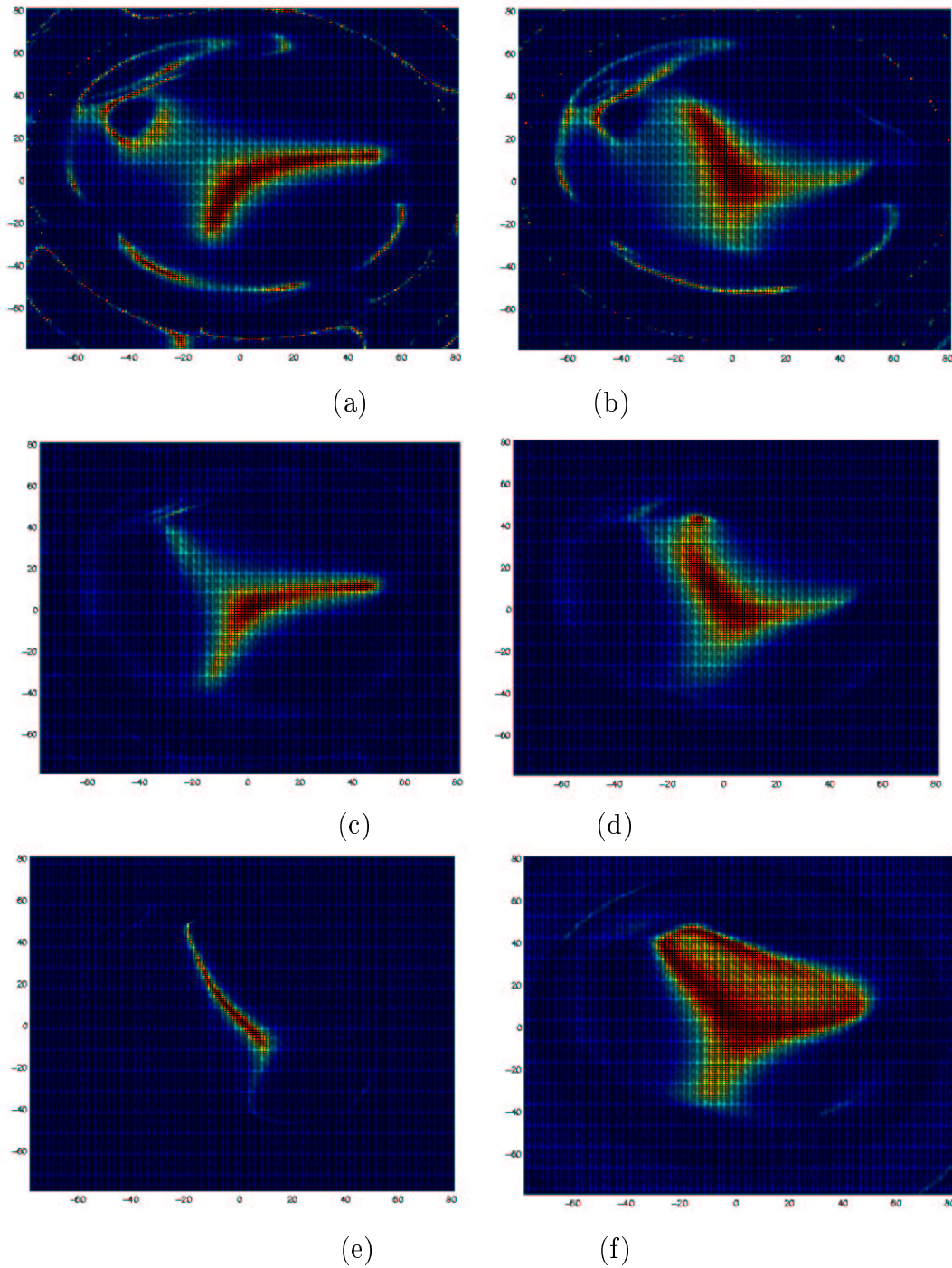


FIG. 5.8 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour le “cerveau” : (a) (r_9, r_4) , (b) (r_9, r_5) , (c) (r_{10}, r_4) , (d) (r_{10}, r_5) , (e) (r_{10}, r_6) , (f) (r_{10}, r_3)

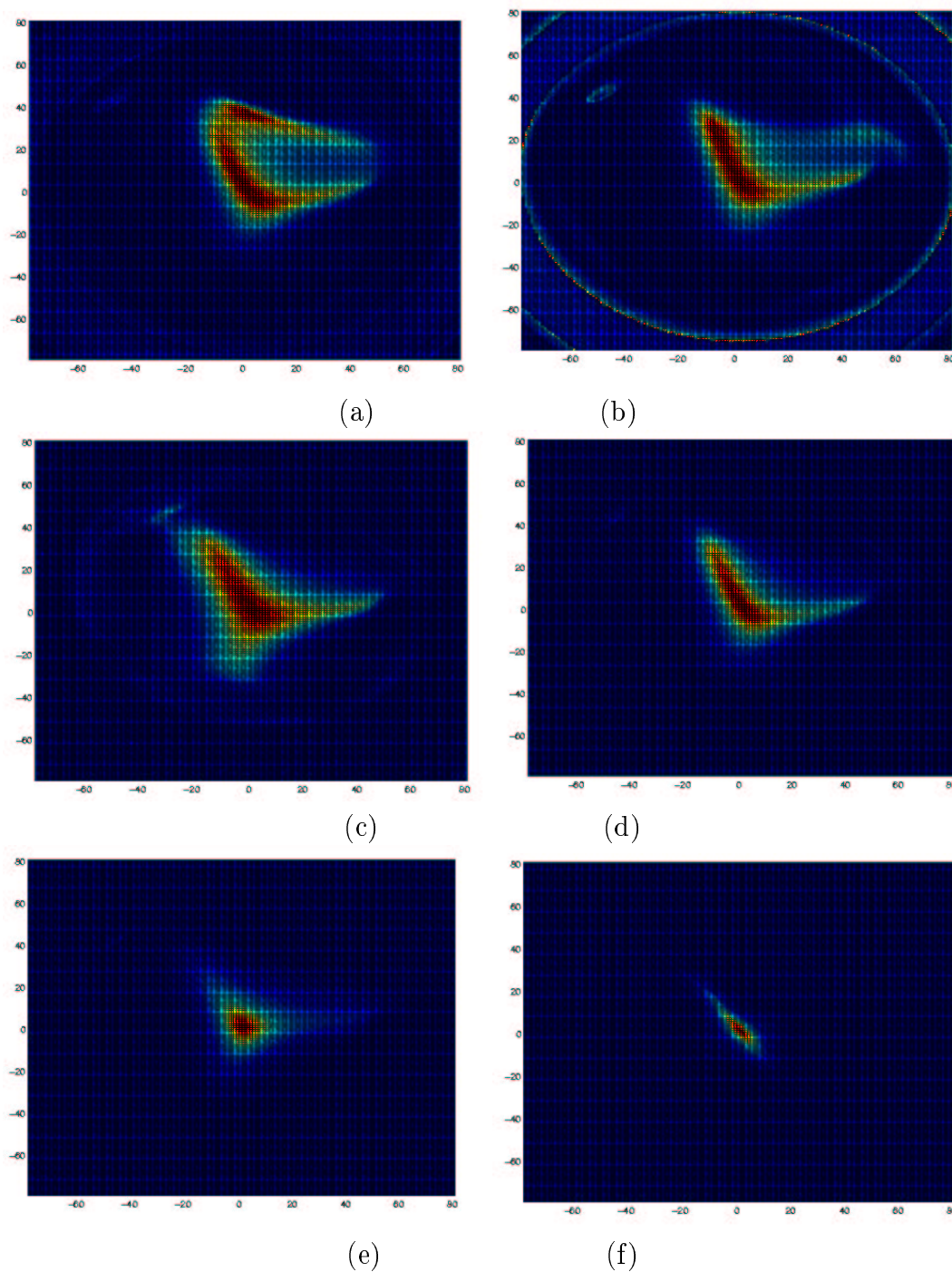


FIG. 5.9 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour le “cerveau” : (a) (r_{10}, r_2) , (b) (r_9, r_2) , (c) (r_9, r_{10}, r_5) , (d) (r_9, r_{10}, r_5, r_2) , (e) $(r_9, r_{10}, r_5, r_4, r_2)$, (f) $(r_9, r_{10}, r_7, r_8, r_6, r_3, r_2, r_4, r_5)$

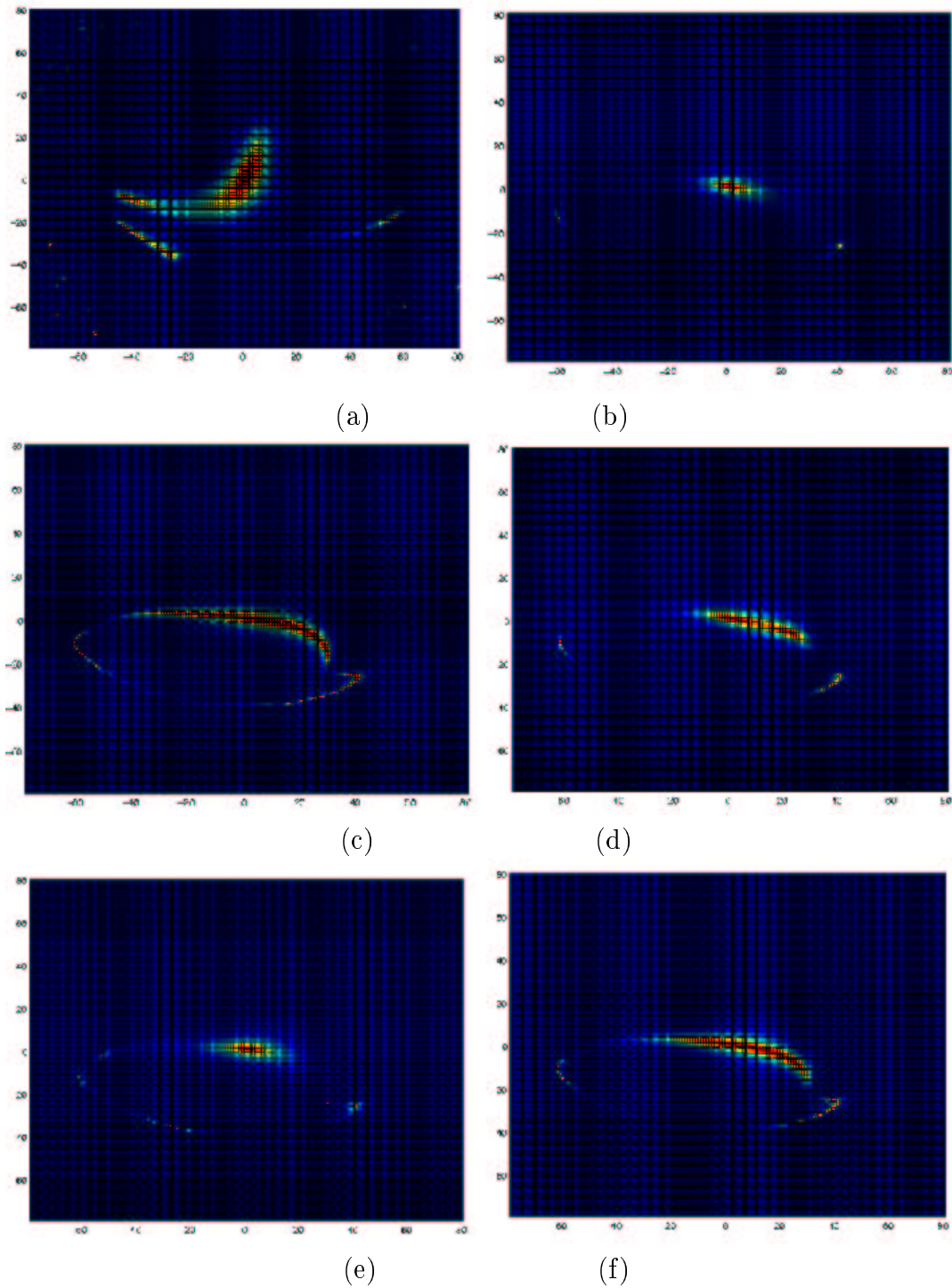


FIG. 5.10 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour la cible de points : (a) (r_9, r_{10}) , (b) $(r_6, r_3, r_2, r_4, r_5)$, (c) (r_6, r_3) , (d) (r_6, r_2) , (e) (r_6, r_4) , (f) (r_6, r_5)

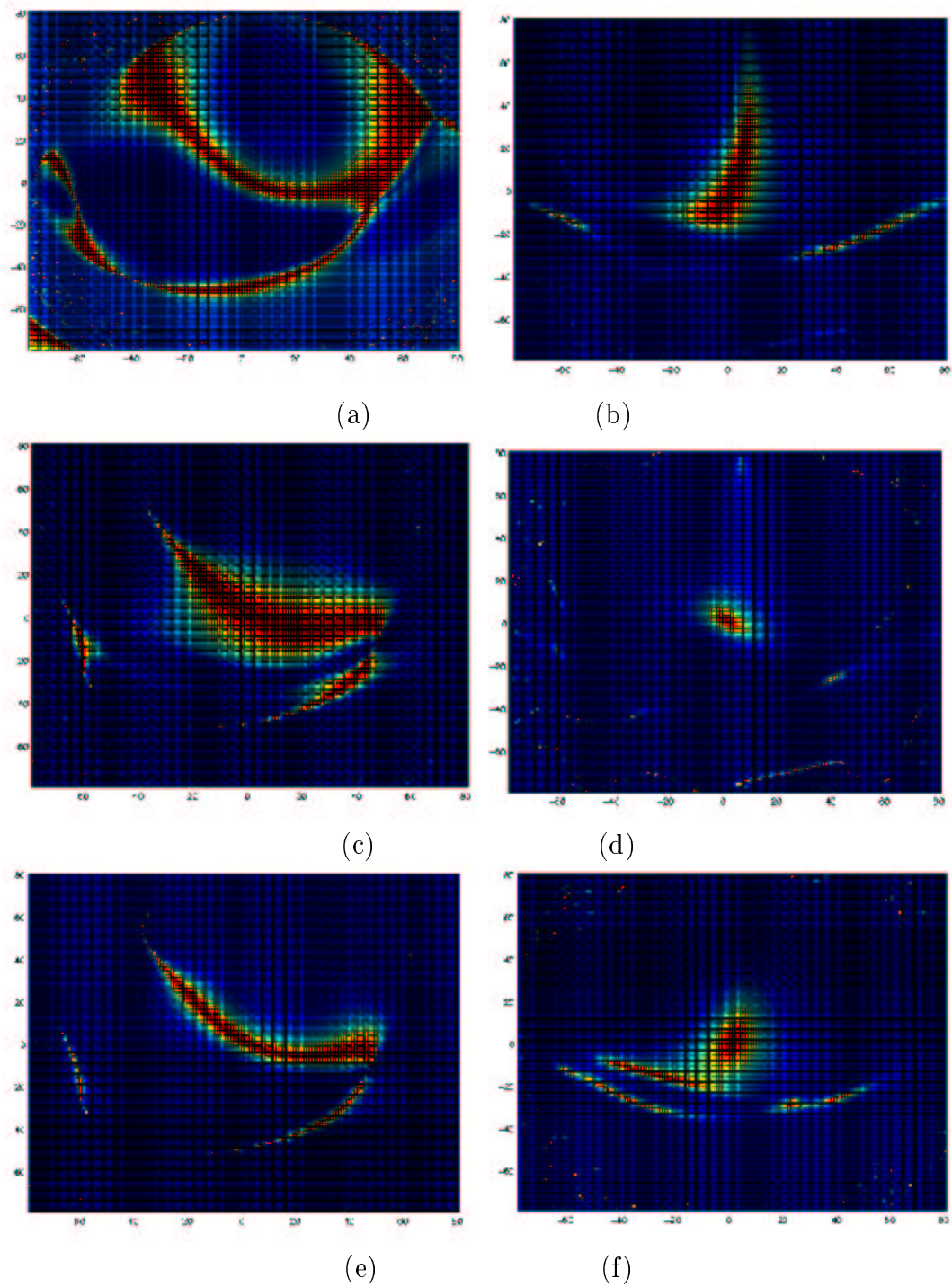


FIG. 5.11 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour la cible de points : (a) (r_3, r_2) , (b) (r_3, r_4) , (c) (r_3, r_5) , (d) (r_2, r_4) , (e) (r_2, r_5) , (f) (r_4, r_5)

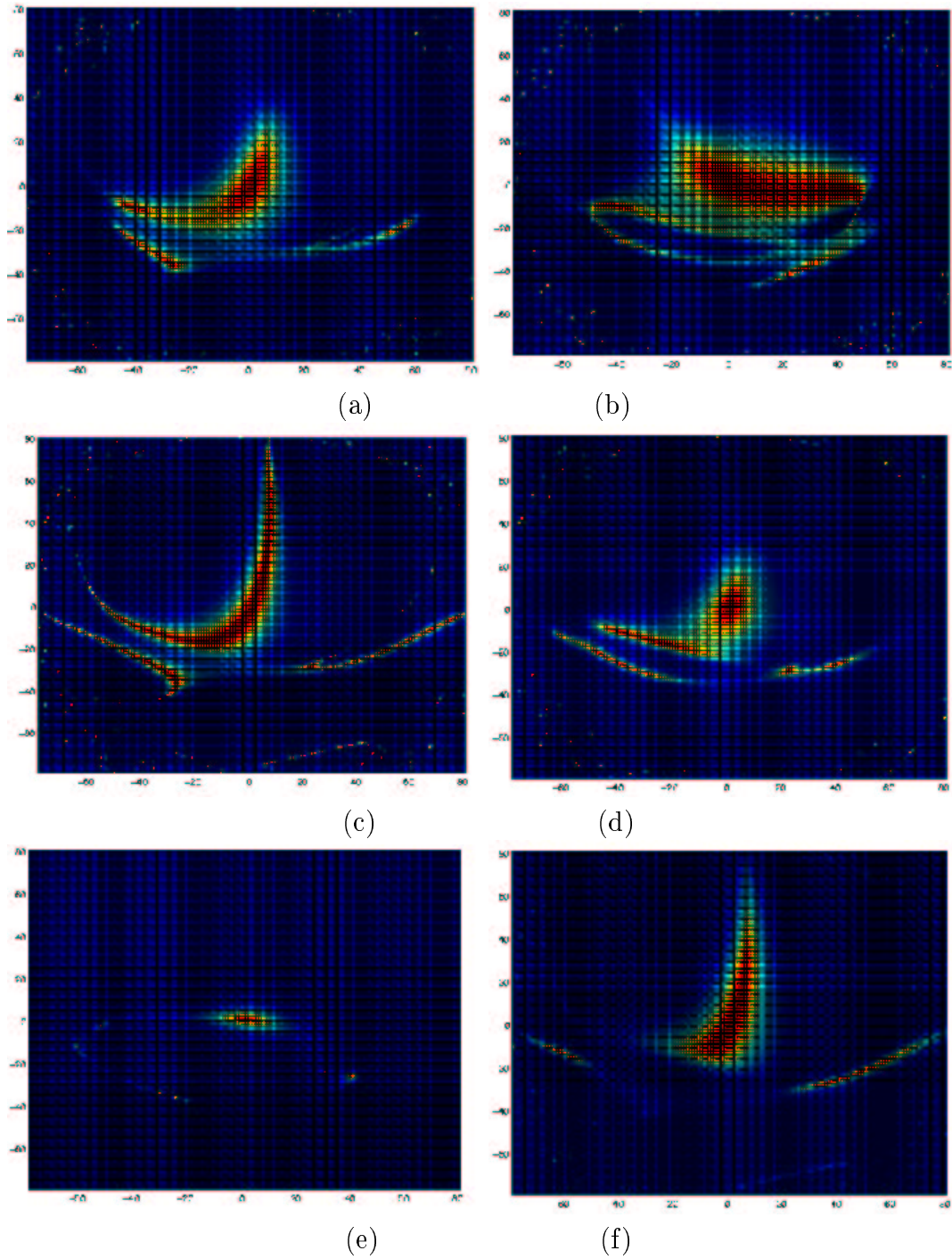


FIG. 5.12 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour la cible de points : (a) (r_9, r_4) , (b) (r_9, r_5) , (c) (r_{10}, r_4) , (d) (r_{10}, r_5) , (e) (r_{10}, r_6) , (f) (r_{10}, r_3)

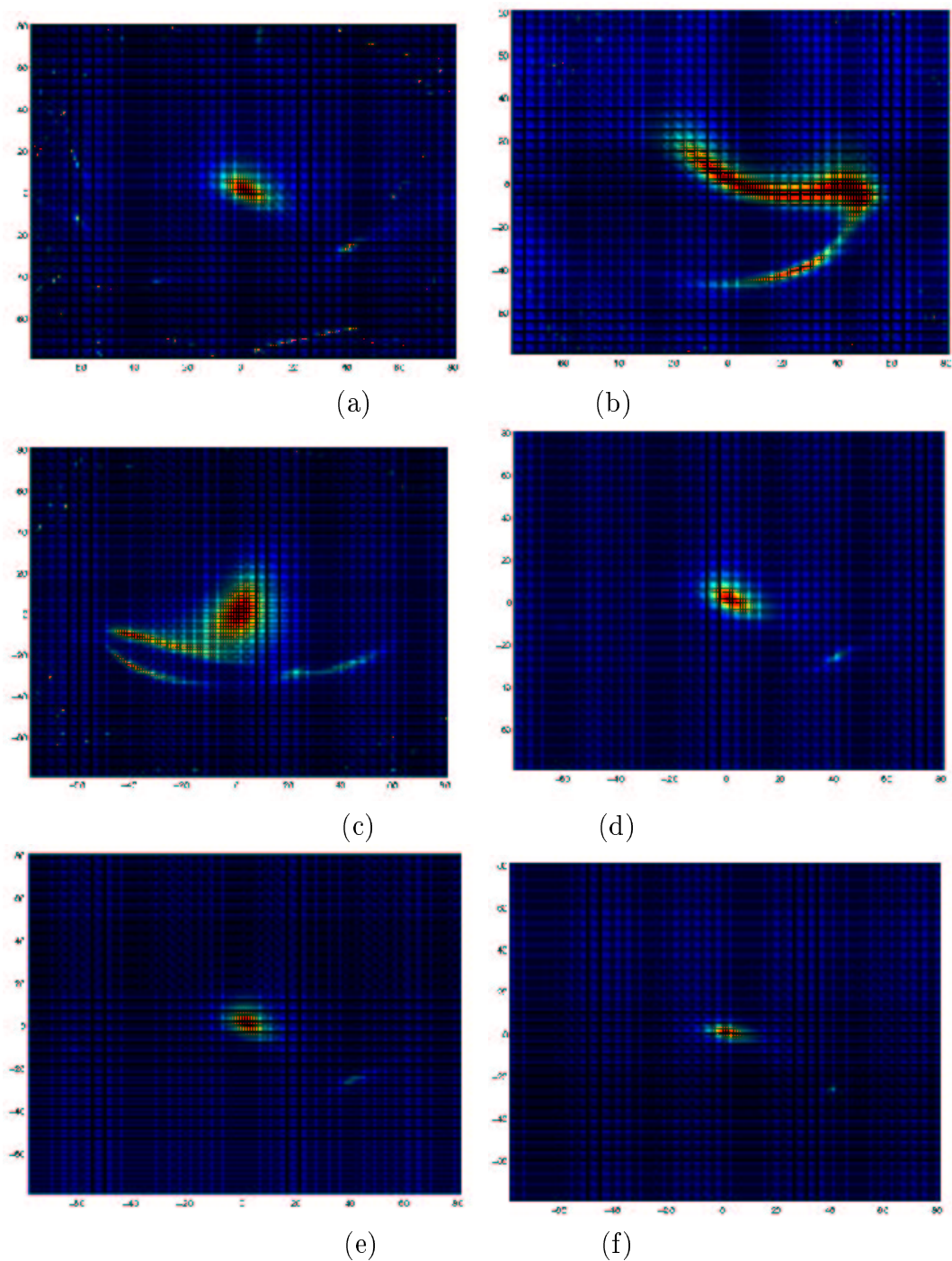


FIG. 5.13 – Représentation de $f(e_r)$ avec des niveaux de couleur selon la combinaison des invariants utilisés obtenus pour la cible de points : (a) (r_{10}, r_2) , (b) (r_9, r_2) , (c) (r_9, r_{10}, r_5) , (d) (r_9, r_{10}, r_5, r_2) , (e) $(r_9, r_{10}, r_5, r_4, r_2)$, (f) $(r_9, r_{10}, r_7, r_8, r_6, r_3, r_2, r_4, r_5)$

5.2 Validation expérimentale : cas d'un objet plan et continu

Dans ce paragraphe, on présente quelques résultats expérimentaux obtenus avec un système robotique à six degrés de liberté. Nous présentons des séries d'expériences en utilisant les deux objets continus donnés par les figures 5.1.a et 5.1.b. Les moments sont calculés à la cadence vidéo après une simple binarisation de l'image sans aucune segmentation spatiale. Pour ces deux séries d'expériences, les paramètres du plan de l'objet dans le repère caméra sont donnés approximativement pour la position désirée, sauf dans le cas où nous considérons des erreurs de modélisation pour tester la robustesse du système. Dans le cas où l'objet est parallèle au plan image pour la position désirée, nous avons fixé ($A = B = 0$, $C = 2$, qui correspond à $Z^* = 0.5$ m). Le cas d'une position désirée de la caméra telle que son plan image ne soit pas parallèle à l'objet est également traité dans ce chapitre. Dans tous les cas, la valeur désirée \mathbf{s}^* est donnée par :

$$\mathbf{s}^* = \left(\widehat{Z}^* x_g^*, \widehat{Z}^* y_g^*, \widehat{Z}^*, r_i^*, r_j^*, \alpha^* \right) \quad (5.2)$$

où x_g^* , y_g^* , r_i^* , r_j^* et α^* sont calculés hors-ligne directement à partir de l'image désirée et où \widehat{Z}^* a été fixé à 0.5 m. On peut signaler que l'utilisation d'une valeur erronée \widehat{Z}^* de Z^* n'a pas d'effet sur la position de convergence du système (on a en effet $\mathbf{s} = \mathbf{s}^*$ seulement pour la position désirée, et ce quelle que soit la valeur \widehat{Z}^*). Une telle erreur induit seulement un effet de gain (égal à \widehat{Z}^*/Z^*) sur la décroissance des trois premières informations visuelles.

Dans la suite de ce paragraphe, nous testons d'abord notre schéma de commande dans le cas où un mouvement de translation pure a été considéré entre la position initiale et la position désirée de la caméra. Nous nous intéressons ensuite au cas où les déplacements à réaliser sont composés de grandes rotations et de translations. Puis nous testons la robustesse de notre méthode vis-à-vis d'erreurs de modélisation.

5.2.1 Mouvement de translation pure

Pour mettre en évidence l'influence de la non linéarité de la matrice d'interaction sur le comportement 3D du robot, nous comparons dans un premier temps les résultats obtenus en utilisant nos informations visuelles et en utilisant les coordonnées du centre de gravité (x_g, y_g) et la surface a dans le cas où le déplacement est une translation pure entre la position initiale et la position désirée de la caméra. Dans un premier temps, la position désirée est telle que le plan image soit approximativement parallèle à l'objet. La valeur de la matrice d'interaction calculée pour l'image désirée ($\widehat{\mathbf{L}}_{\mathbf{s}} = \mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}$) a été utilisée dans la loi de commande (3.18) et la valeur du gain λ a été fixée à 0.1.

À partir de la figure 5.14, on peut noter les larges améliorations apportées par l'utilisation des informations visuelles proposées dans cette thèse (courbes en poin-

tillé). En effet, on obtient la même décroissance exponentielle découplée pour les informations visuelles dans l'image et pour les composantes de la vitesse de la caméra. Comme prévu, la trajectoire obtenue en utilisant les informations visuelles proposées est rectiligne, ce qui n'est pas le cas en utilisant les autres informations visuelles.

La figure 5.15 donne les résultats obtenus dans les mêmes conditions (c'est-à-dire un mouvement de translation pure) en utilisant comme objet une image de "cerveau". Ces résultats confirment ceux obtenus pour la "baleine". En effet, on obtient la même décroissance exponentielle découplée pour les informations visuelles dans l'image et pour les composantes des vitesses de translation et une trajectoire rectiligne entre les positions initiale et désirée de la caméra.

Le même schéma de commande a ensuite été testé dans le cas où le plan image de la caméra dans sa position désirée est non parallèle à l'objet. Les figures 5.16 et 5.17 donnent respectivement les résultats obtenus pour "la baleine" et "le cerveau" dans ce cas de figure. La rotation virtuelle à appliquer sur la caméra dans sa position désirée pour se ramener au cas parallèle a été fixée dans les deux cas à 30 degrés autour de l'axe x de la caméra. À partir des figures 5.16 et 5.17, on note que les bonnes propriétés obtenues dans le cas où le plan image de la caméra dans sa position désirée est parallèle à l'objet restent valables dans le cas non parallèle.

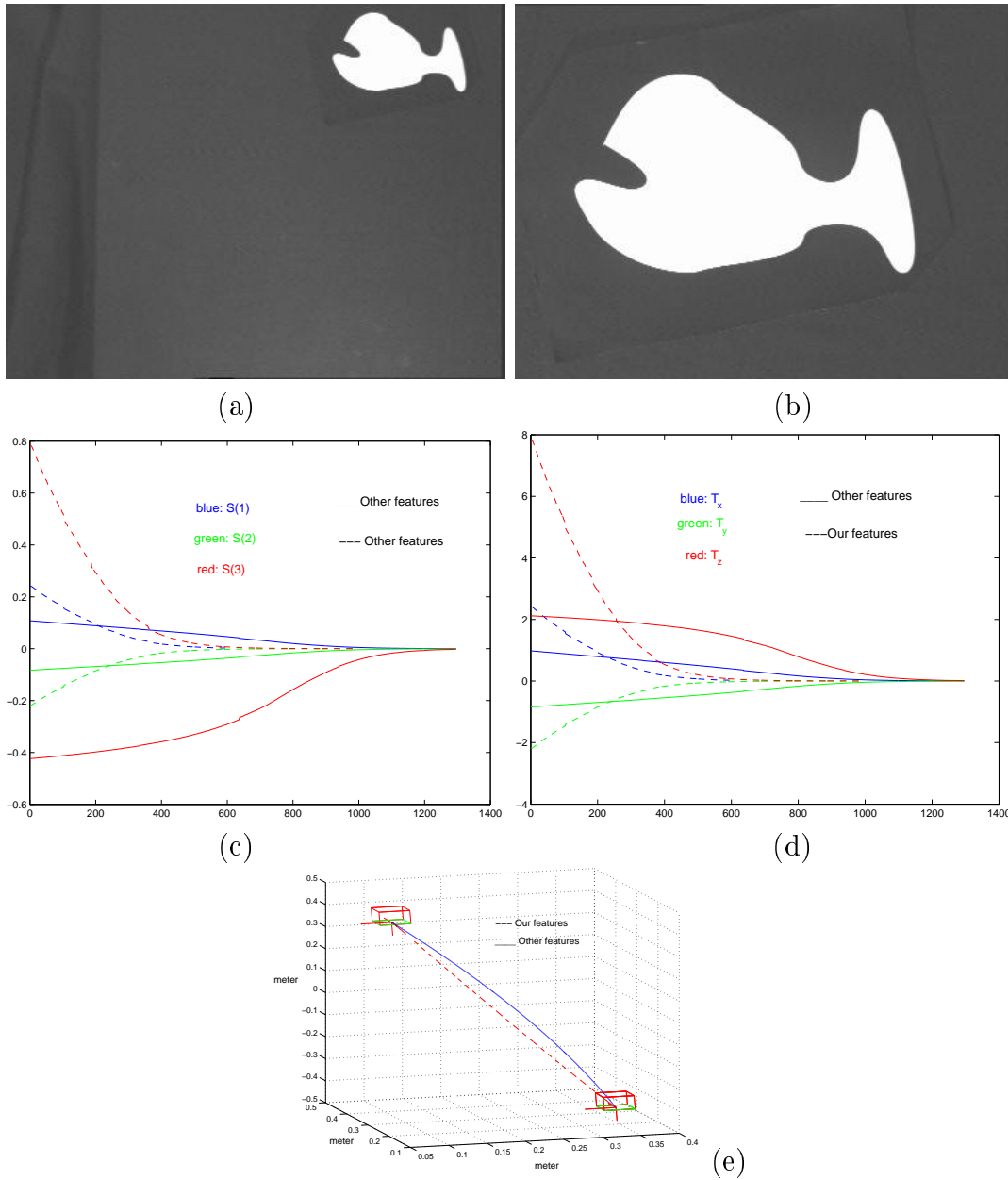


FIG. 5.14 – Résultats pour un mouvement de translation pure pour une position désirée de la caméra où son plan image est parallèle à l'objet (baleine): (a) image initiale, (b) image désirée, (c) informations visuelles $(\mathbf{s} - \mathbf{s}^*)$, (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

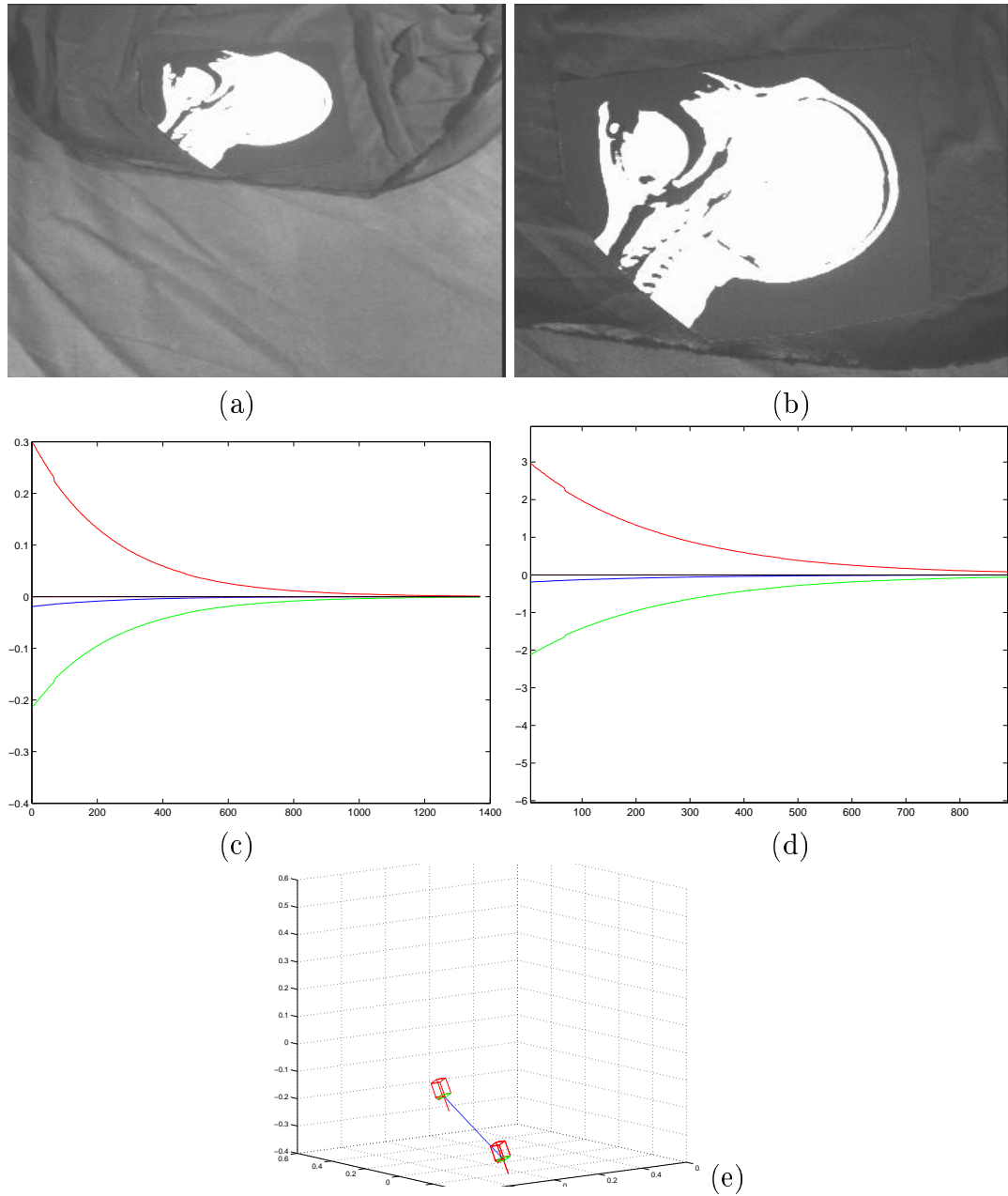


FIG. 5.15 – Résultats pour un mouvement de translation pure pour une position désirée de la caméra où son plan image est parallèle à l'objet (cerveau): (a) image initiale, (b) image désirée, (c) informations visuelles ($\mathbf{s} - \mathbf{s}^*$), (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

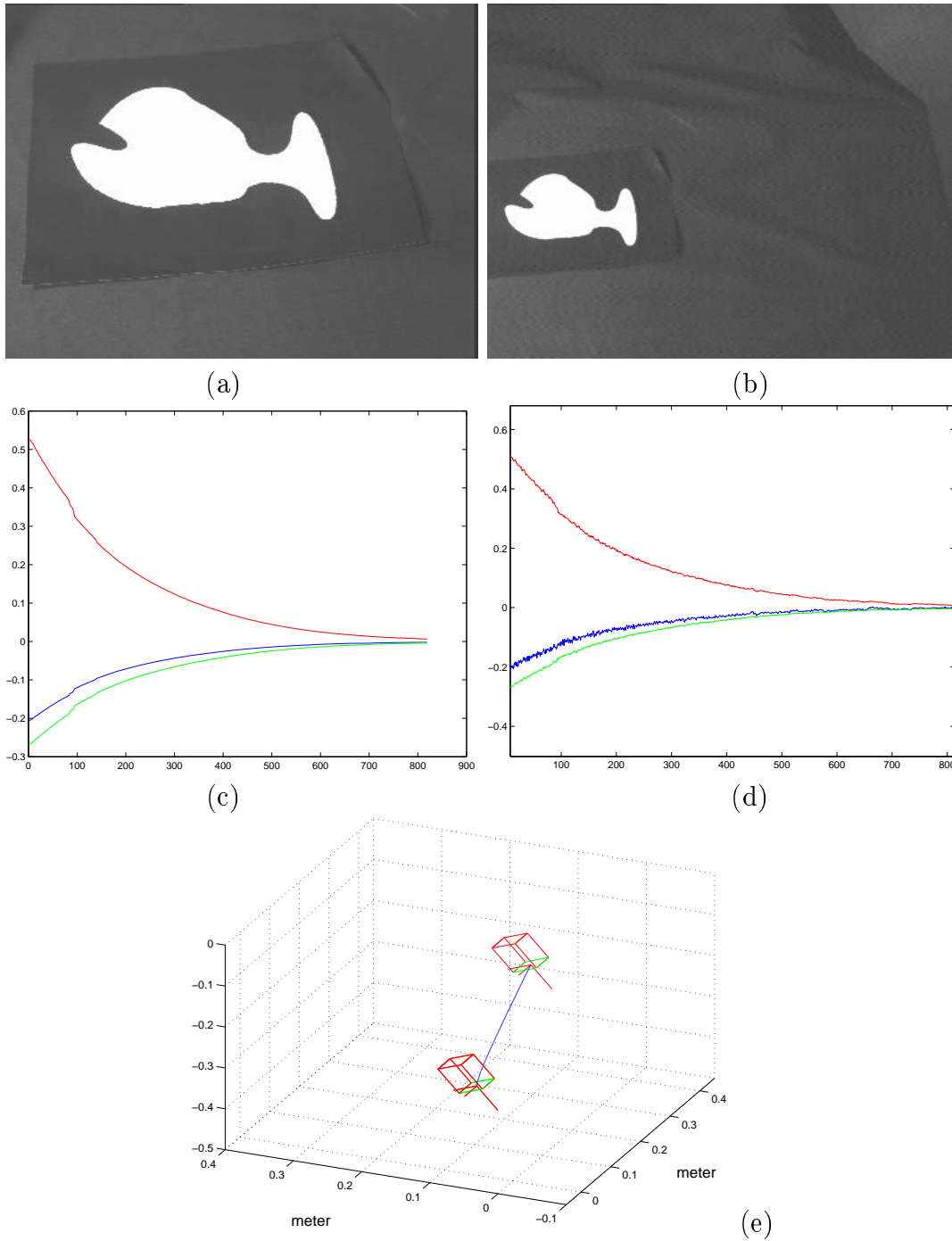


FIG. 5.16 – Résultats pour un mouvement de translation pure pour une position désirée de la caméra où son plan image n'est pas parallèle à l'objet (baleine): (a) image initiale, (b) image désirée, (c) informations visuelles ($\mathbf{s} - \mathbf{s}^*$), (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

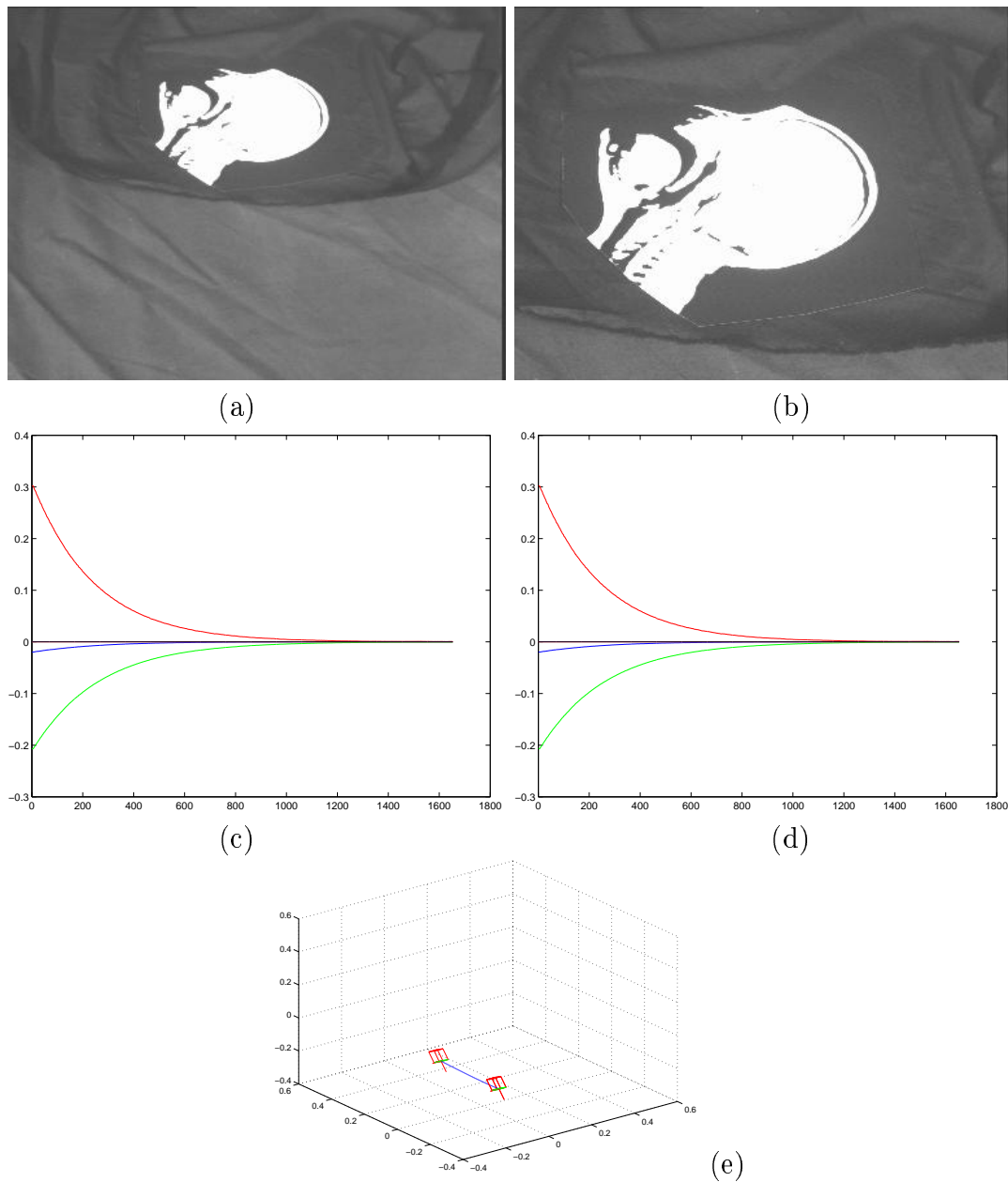


FIG. 5.17 – Résultats pour un mouvement de translation pure pour une position désirée de la caméra où son plan image n'est pas parallèle à l'objet (cerveau) : (a) image initiale, (b) image désirée, (c) informations visuelles ($\mathbf{s} - \mathbf{s}^*$), (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

5.2.2 Mouvement complexe

On teste maintenant notre schéma de commande pour des déplacements à réaliser comportant des grandes translations et rotations. Comme dans la série d'expérimentation précédente, nous considérons deux cas de figures d'orientation du plan image par rapport à l'objet dans sa position désirée. Dans un premier temps, nous considérons le cas où les plans image et celui de l'objet sont parallèles pour la position désirée. Nous utiliserons également les deux mêmes objets ("baleine" et "cerveau"). Pour contrôler les vitesses de rotation ω_x et ω_y , nous avons choisi le couple d'invariants (r_{10}, r_9) dans le cas de la "baleine" et le couple (r_6, r_4) dans le cas du "cerveau". Les valeurs numériques de la matrice d'interaction calculée pour la position désirée pour chaque objet sont respectivement :

$$\mathbf{L}_{s|s=s^*}^{\parallel} = \begin{pmatrix} -1 & 0 & 0 & 0.01 & -0.52 & 0.01 \\ 0 & -1 & 0 & 0.51 & -0.01 & 0.01 \\ 0 & 0 & -1 & -0.02 & -0.01 & 0 \\ 0 & 0 & 0 & -0.61 & 0.09 & 0 \\ 0 & 0 & 0 & -0.33 & -0.62 & 0 \\ 0 & 0 & 0 & -0.04 & -0.08 & -1 \end{pmatrix} \quad (5.3)$$

et

$$\mathbf{L}_{s|s=s^*}^{\parallel} = \begin{pmatrix} -1 & 0 & 0 & -0.00 & -0.51 & 0.00 \\ 0 & -1 & 0 & 0.51 & 0.00 & 0.01 \\ 0 & 0 & -1 & -0.00 & -0.02 & 0 \\ 0 & 0 & 0 & -3.38 & -2.68 & 0 \\ 0 & 0 & 0 & -0.90 & 1.03 & 0 \\ 0 & 0 & 0 & -0.10 & 0.00 & -1 \end{pmatrix} \quad (5.4)$$

On peut noter que ces matrices sont bloc triangulaires avec des valeurs non nulles principalement sur la diagonale. Les valeurs du conditionnement de cette matrice (égal à 2.60 dans le cas de la "baleine" et à 3.14 dans le cas du "cerveau") sont également très satisfaisantes. Pour cette série d'expériences, nous avons utilisé dans la loi de commande :

$$\hat{\mathbf{L}}_s = \frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel}) \quad (5.5)$$

c'est-à-dire que la pose n'est pas estimée au cours de l'asservissement.

Les résultats obtenus sont donnés sur les figures 5.18, 5.19 et 5.20. Ils montrent le bon comportement de la loi de commande. D'abord, nous pouvons noter la convergence rapide vers la position désirée. De plus, la trajectoire 3D de la caméra obtenue dans chacun des cas considérés (voir figures 5.18.e., 5.19.e et 5.20.e) est très satisfaisante même si le déplacement à réaliser était très grand et que les paramètres courants du plan objet dans le repère caméra ne sont pas estimés.

Nous présentons ensuite des résultats expérimentaux obtenus dans le cas d'une position désirée de la caméra telle que son plan image ne soit pas parallèle à l'objet.

Les positions désirées de la caméra sont les mêmes que celles utilisées dans le cas où le mouvement à effectuer est une translation pure (soit une rotation de 30 degrés autour de l'axe x de la caméra). Nous avons choisi les mêmes couples d'invariants utilisés dans le cas où l'objet et le plan image sont parallèles pour la position désirée de la caméra. Les résultats obtenus sont donnés sur les figures 5.21, 5.22 et 5.23. Ces résultats montrent que le bon découplage dans le cas parallèle reste valable dans le cas non parallèle. Ainsi, une décroissance exponentielle des vitesses a été obtenue dans la majorité des cas, excepté des petites oscillations pour certaines vitesses. Les trajectoires de la caméra sont également satisfaisantes dans chacun des cas présentés. Les valeurs numériques de la matrice d'interaction pour les positions désirées après application de la rotation virtuelle sont données par les équations (5.6) et (5.7). Ces matrices présentent un bon conditionnement (égal à 7.8 et 5.33).

$$\mathbf{L}_{\mathbf{s}|\mathbf{s}^*}^{\parallel} = \begin{pmatrix} -1 & 0 & 0 & 0.01 & -0.52 & -0.37 \\ 0 & -1 & 0 & 0.36 & -0.00 & 0.01 \\ 0 & 0 & -1 & 0.56 & -0.02 & 0 \\ 0 & 0 & 0 & -1.10 & -3.20 & 0 \\ 0 & 0 & 0 & 0.24 & -3.32 & 0 \\ 0 & 0 & 0 & -0.42 & 0.93 & -1 \end{pmatrix} \quad (5.6)$$

$$\mathbf{L}_{\mathbf{s}|\mathbf{s}^*}^{\parallel} = \begin{pmatrix} -1 & 0 & 0 & -0.01 & -0.51 & -0.18 \\ 0 & -1 & 0 & 0.47 & 0.01 & 0.04 \\ 0 & 0 & -1 & 0.27 & -0.06 & 0 \\ 0 & 0 & 0 & -1.74 & -3.70 & 0 \\ 0 & 0 & 0 & -0.54 & 2.36 & 0 \\ 0 & 0 & 0 & 0.15 & 0.66 & -1 \end{pmatrix} \quad (5.7)$$

5.2.3 Résultats avec une mauvaise calibration de la caméra et une occultation de l'objet

Nous examinons maintenant la robustesse de notre approche vis-à-vis d'une mauvaise calibration du système. Pour cela, des erreurs ont été ajoutées sur les paramètres intrinsèques de la caméra (25% sur la focale et 20 pixels sur les coordonnées du point principal) ainsi que sur les paramètres du plan objet ($\hat{Z}^* = 0.8m$ à la place de $Z^* = 0.5m$). Les conditions d'éclairage entre les positions initiale et désirée sont également différentes (voir Figure 5.24.a et 5.24.b). Par ailleurs, une occultation d'une partie de l'objet dans la position initiale a été introduite volontairement. Les résultats obtenus sont donnés sur la figure 5.24. Nous pouvons noter que le système converge malgré les mauvaises conditions d'expérimentations. Une fois l'occultation terminée, le comportement du système devient similaire à celui obtenu dans les expérimentations précédentes, ce qui valide la robustesse de notre méthode vis-à-vis de ces erreurs.

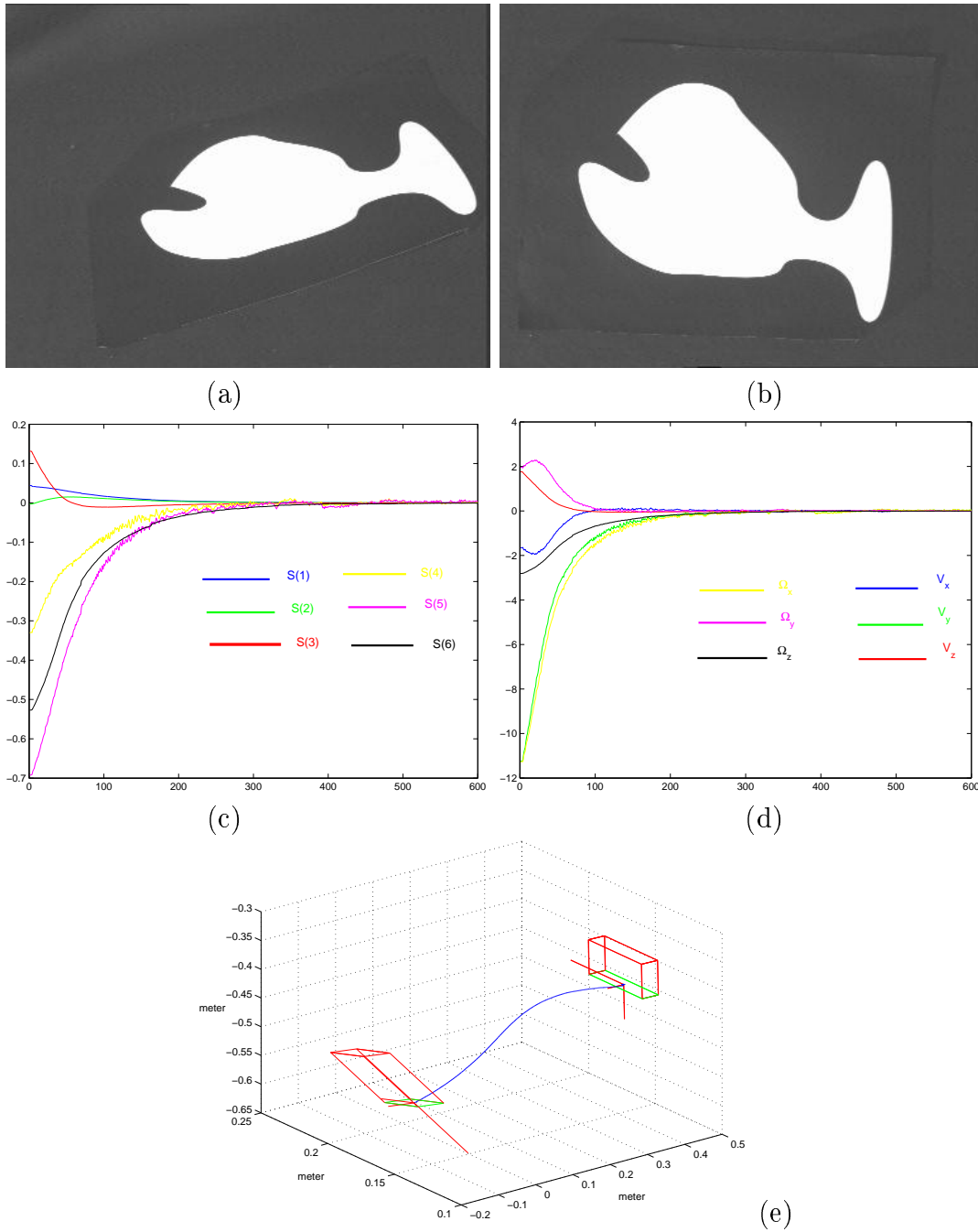


FIG. 5.18 – Résultats pour un mouvement complexe obtenus pour la “baleine” pour une position de la caméra où son plan image n’est pas parallèle à l’objet : (a) image initiale, (b) image désirée, (c) informations visuelles ($\mathbf{s} - \mathbf{s}^*$), (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

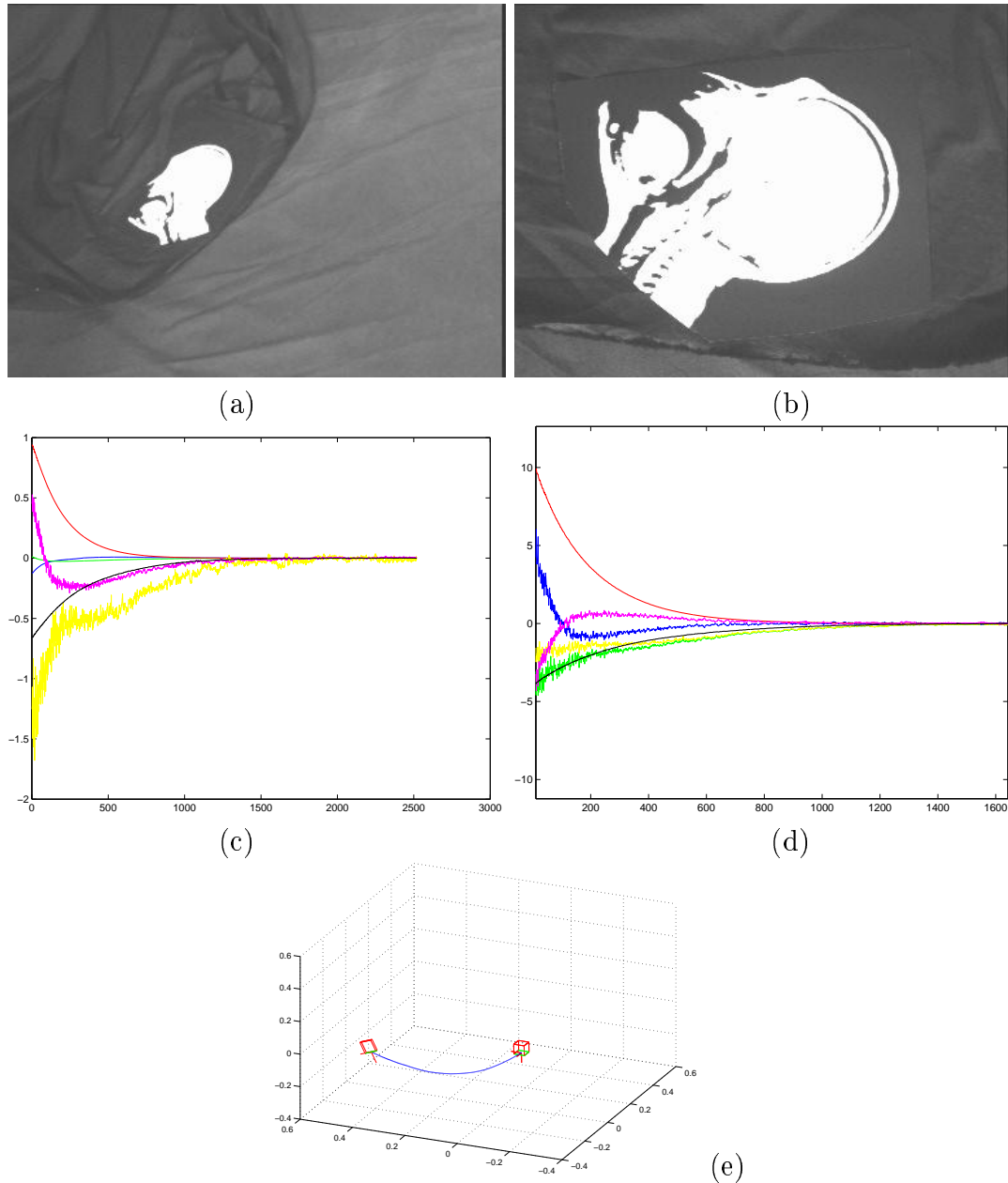


FIG. 5.19 – Résultats pour un mouvement complexe pour une position désirée de la caméra où son plan image est parallèle à l'objet (cerveau): (a) image initiale, (b) image désirée, (c) informations visuelles $(s - s^*)$, (d) vitesses de la caméra v_c , (e) trajectoire 3D de la caméra

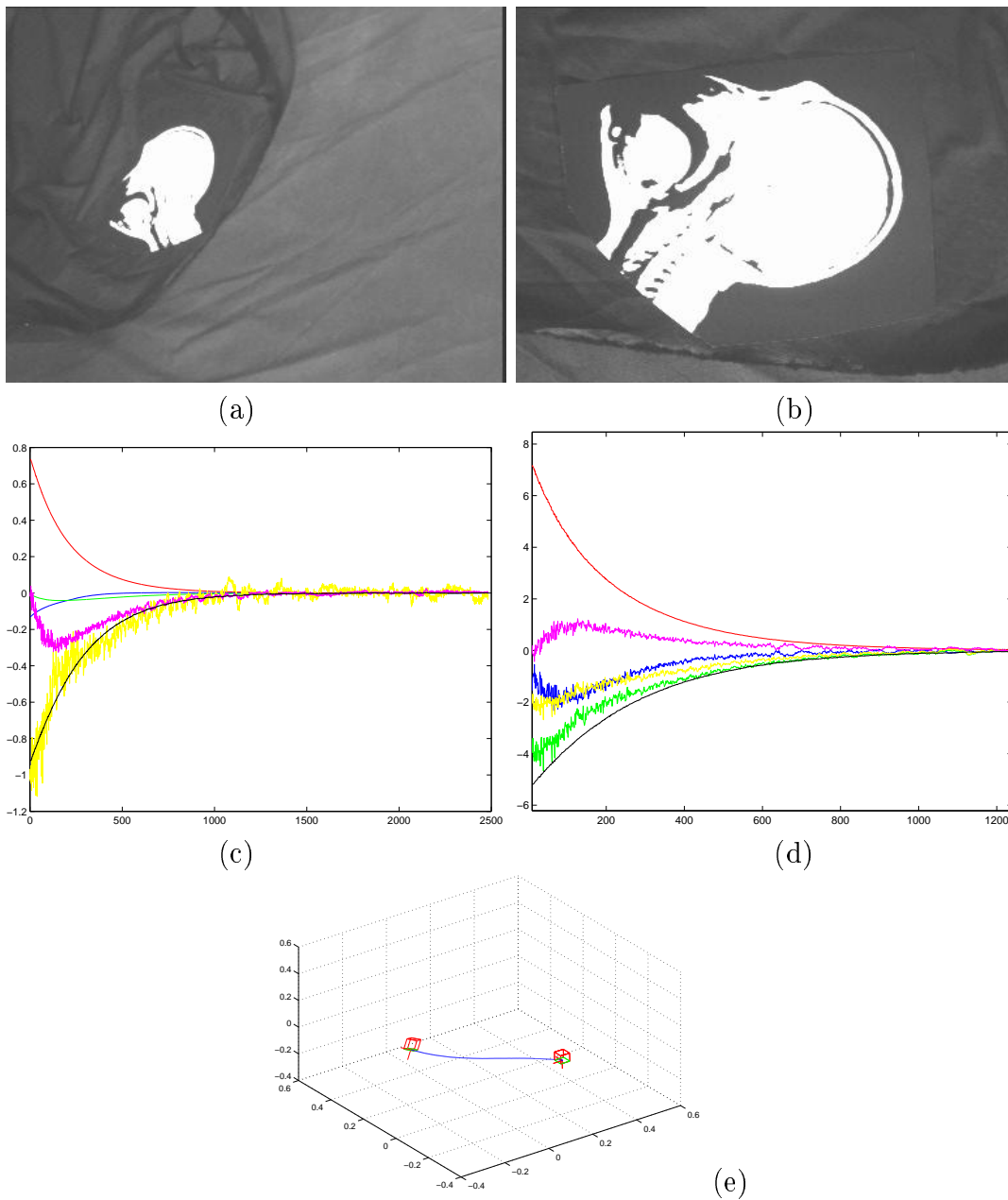


FIG. 5.20 – Résultats pour un mouvement complexe pour une position désirée de la caméra où son plan image est parallèle à l'objet (cerveau): (a) image initiale, (b) image désirée, (c) informations visuelles $(\mathbf{s} - \mathbf{s}^*)$, (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

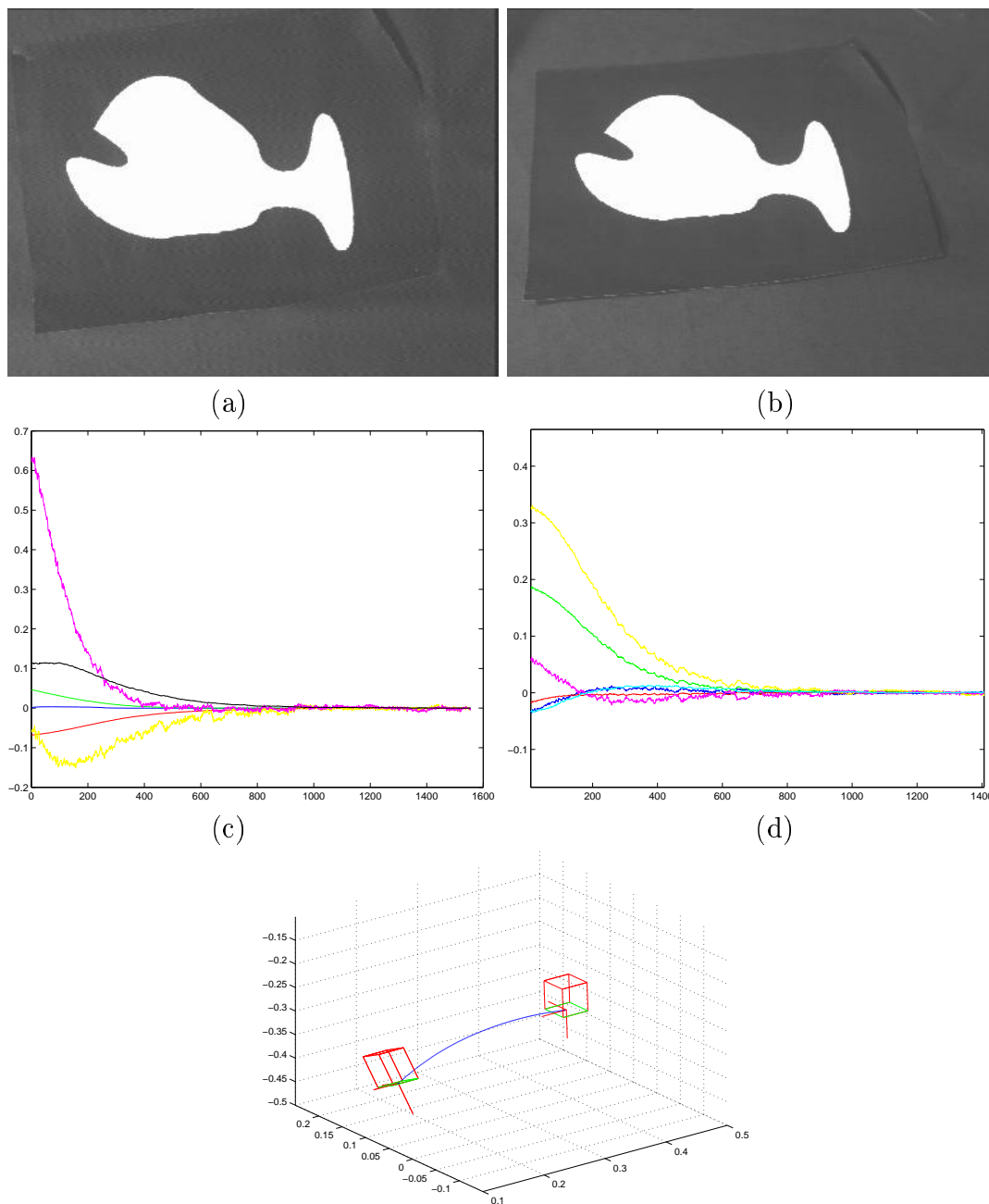


FIG. 5.21 – Résultats pour un mouvement complexe et une position désirée de la caméra non parallèle au plan image: (a) image initiale, (b) image désirée, (c) erreur moyenne sur les informations visuelles $(\mathbf{s} - \mathbf{s}^*)$, (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

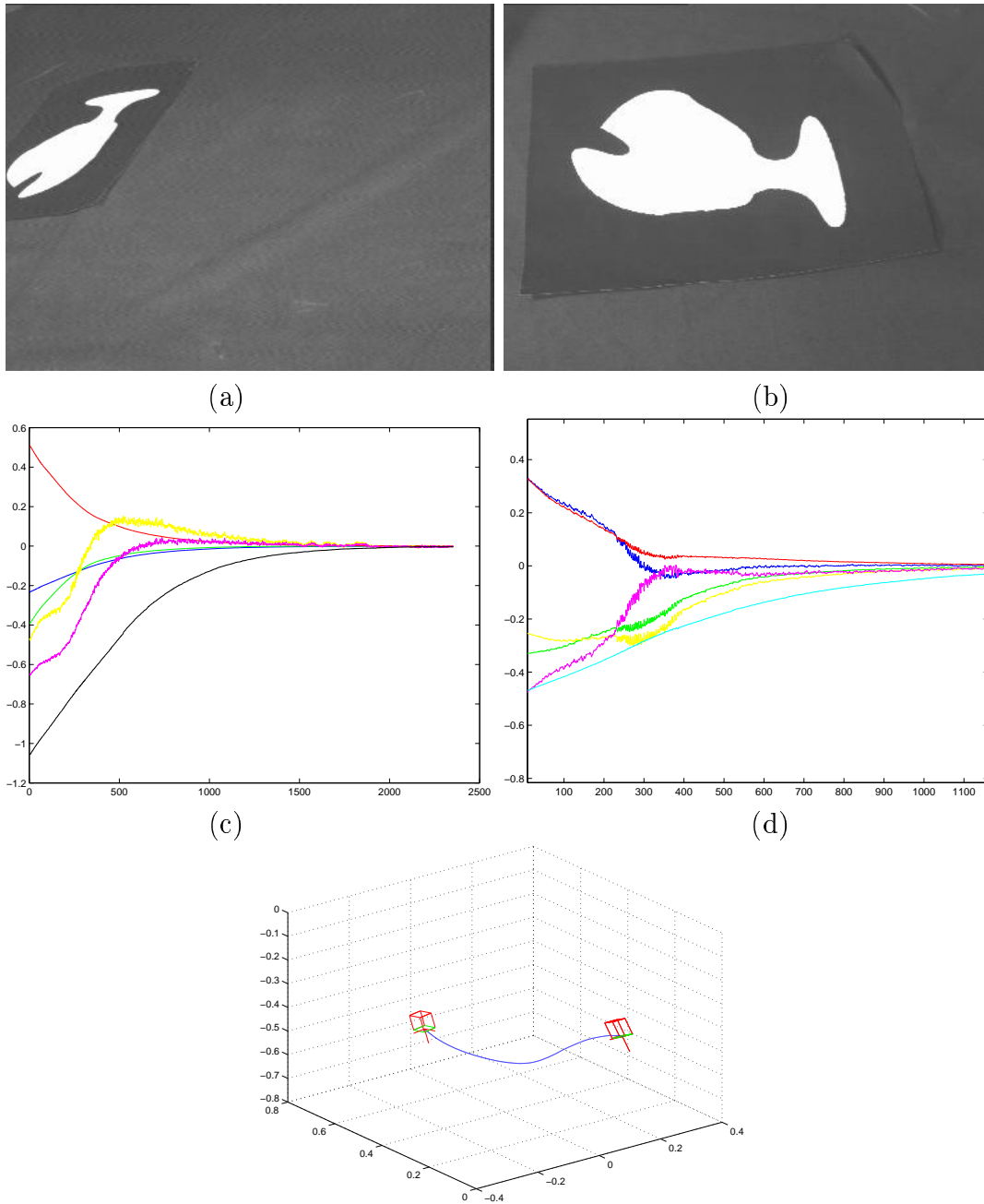


FIG. 5.22 – Résultats pour un mouvement complexe et une position désirée de la caméra non parallèle au plan image: (a) image initiale, (b) image désirée, (c) erreur moyenne sur les informations visuelles $(\mathbf{s} - \mathbf{s}^*)$, (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

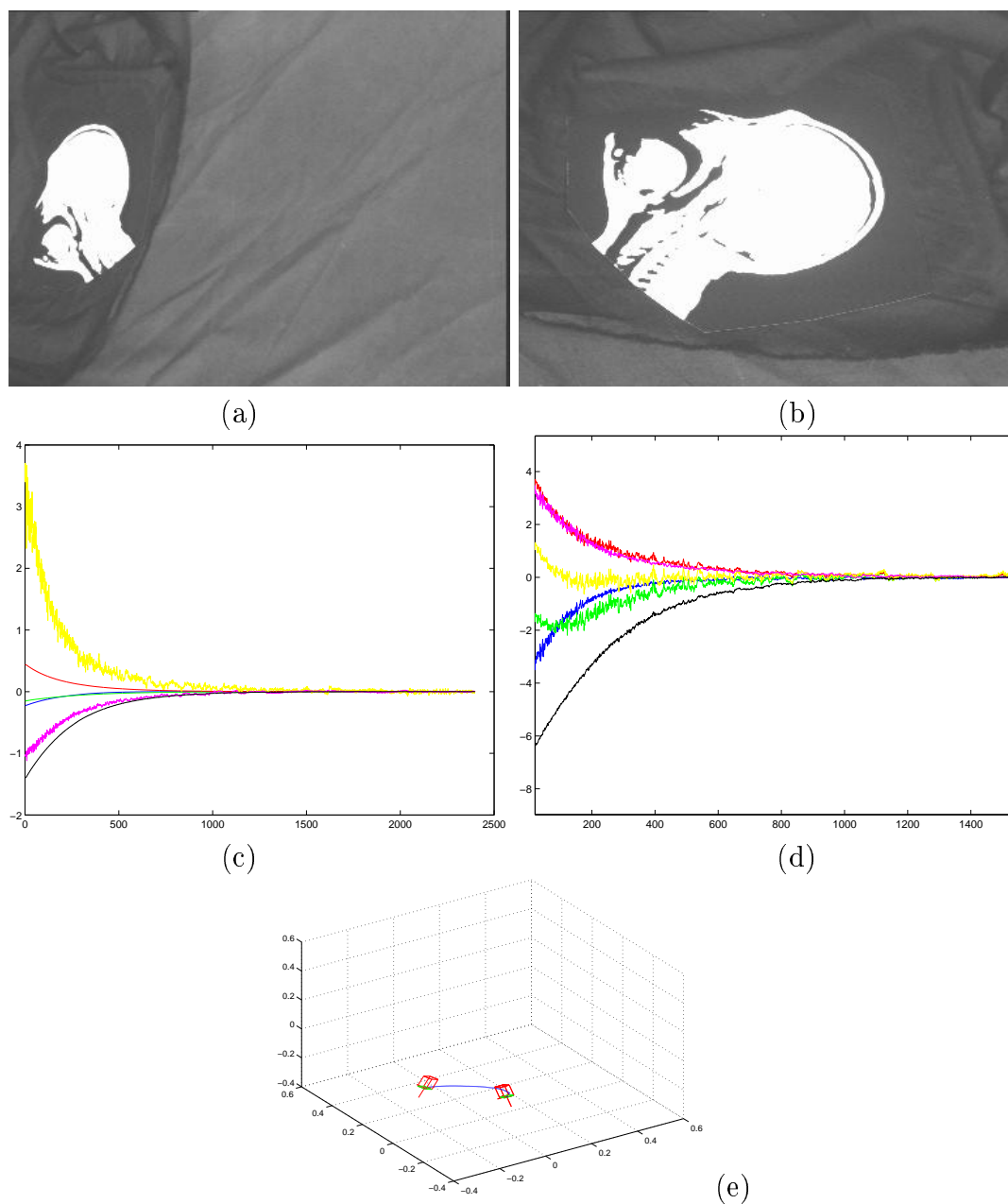


FIG. 5.23 – Résultats pour un mouvement complexe pour une position désirée de la caméra où son plan image n'est pas parallèle à l'objet (cerveau): (a) image initiale, (b) image désirée, (c) informations visuelles $(\mathbf{s} - \mathbf{s}^*)$, (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

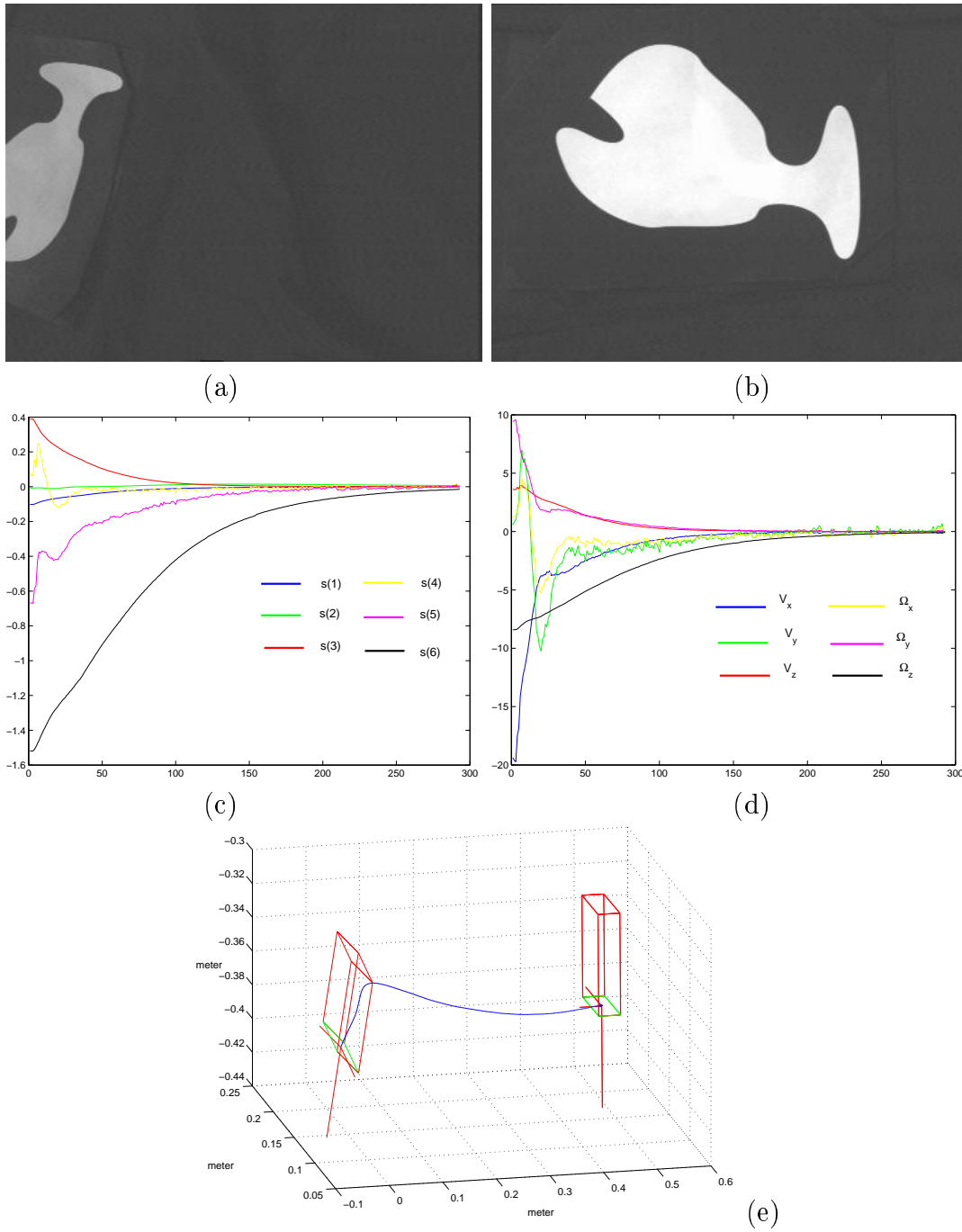


FIG. 5.24 – Résultats avec une mauvaise calibration et une occultation initiale: (a) image initiale, (b) image désirée (c) informations visuelles $(\mathbf{s} - \mathbf{s}^*)$, (d) vitesses de la caméra \mathbf{v}_c , (e) trajectoire 3D de la caméra

5.2.4 Cas d'un objet symétrique : un rectangle

Dans ce paragraphe, nous considérons un objet symétrique dont les coins peuvent être facilement suivis dans les images acquises. Les moments sont calculés à partir des coordonnées des quatre sommets du quadrilatère observé dans l'image. La position désirée de la caméra est tel que le rectangle soit parallèle au plan image et à une distance de 50 cm du centre optique de la caméra ($A = B = 0, C = 2$) et tel que le rectangle soit centré et horizontal dans l'image (voir figure 5.25.a). Le déplacement de la caméra à réaliser est approximativement composé d'une rotation de $-35, -20$ et -65 degrés respectivement autour des axes x, y et z et d'une translation de 10, -25 et 15 cm le long des ces axes. L'image acquise à la position initiale de la caméra est donnée sur la figure 5.25.b.

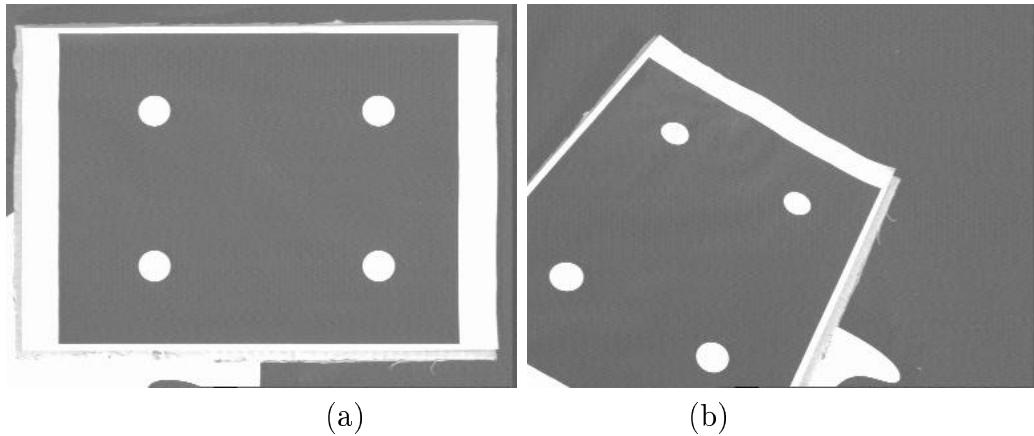


FIG. 5.25 – (a) Image désirée, (b) Image initiale

La valeur de la matrice d'interaction calculée pour la position désirée en utilisant les coordonnées x_i et y_i des coins du rectangle comme informations visuelles est donnée par :

$$\mathbf{L}_{s|s=s^*} = \begin{bmatrix} -2 & 0 & -0.3 & 0.01 & -1.02 & -0.10 \\ 0 & -2 & -0.2 & 1.01 & -0.01 & 0.14 \\ -2 & 0 & 0.3 & -0.01 & -1.02 & -0.10 \\ 0 & -2 & -0.2 & 1.01 & 0.01 & -0.14 \\ -2 & 0 & 0.3 & 0.01 & -1.02 & 0.10 \\ 0 & -2 & 0.2 & 1.01 & -0.01 & -0.14 \\ -2 & 0 & -0.3 & -0.01 & -1.02 & 0.10 \\ 0 & -2 & 0.2 & 1.01 & 0.01 & 0.14 \end{bmatrix}$$

Cette matrice ne présente aucune propriété intéressante de découplage. On peut noter également que la valeur de son conditionnement est approximativement égal à 180. Les résultats obtenus en utilisant les coordonnées de points sont donnés par la figure 5.27. Les points dans l'image convergent vers leur position désirée. Cependant, leur trajectoire ainsi que le comportement des vitesses de la caméra ne

sont pas particulièrement intéressants. On donne à présent les résultats obtenus pour exactement la même expérience mais en utilisant les informations visuelles proposées dans [Chaumette 04] :

$$\mathbf{s} = (x_g, y_g, a, s_x, s_y, \alpha)$$

où les formules de s_x et s_y sont données dans le chapitre 3. La valeur désirée de \mathbf{s}^* est donnée par :

$$\mathbf{s}^* = (0, 0, a^*, 0, 0, 0)$$

La valeur de la matrice d'interaction calculée pour la position désirée de la caméra présente de bonnes propriétés de découplage. Plus précisément, on a :

$$\mathbf{L}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*} = \begin{bmatrix} -2 & 0 & 0 & 0 & -1.02 & 0 \\ 0 & -2 & 0 & 1.01 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.19 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Cette matrice présente un conditionnement égal à 17.1, soit dix fois moins que celui obtenu en utilisant les coordonnées de points comme informations visuelles. Les résultats obtenus sont donnés sur les figures 5.26 et 5.28. En comparant les figures 5.27 et 5.28, on peut noter immédiatement les améliorations apportées aux trajectoires des coins du rectangle et aux vitesses de la caméra. Ces résultats peuvent encore être améliorés en remplaçant les trois premières informations visuelles proposées dans [Chaumette 04] (x_g , y_g et a) par les trois premières informations visuelles proposées dans ce manuscrit (x_n , y_n et a_n).

Dans le paragraphe suivant, nous présentons des résultats expérimentaux obtenus en injectant les paramètres estimés du plan de l'objet dans le calcul de la valeur de la matrice d'interaction. Cette estimation est réalisée en utilisant la méthode de calcul de pose que nous avons proposée dans le chapitre précédent. Nous comparons à cette occasion l'utilisation de plusieurs approximations possibles de la matrice d'interaction dans la boucle de commande.

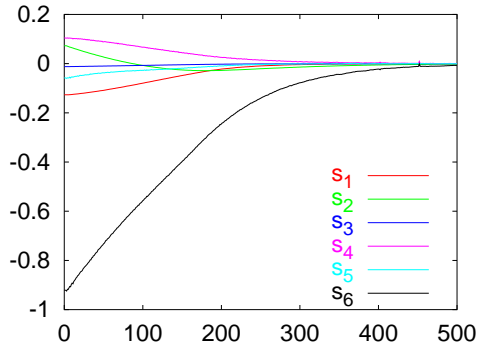


FIG. 5.26 – Résultats obtenus en utilisant les moments comme informations visuelles : Erreurs sur les informations visuelles (m)

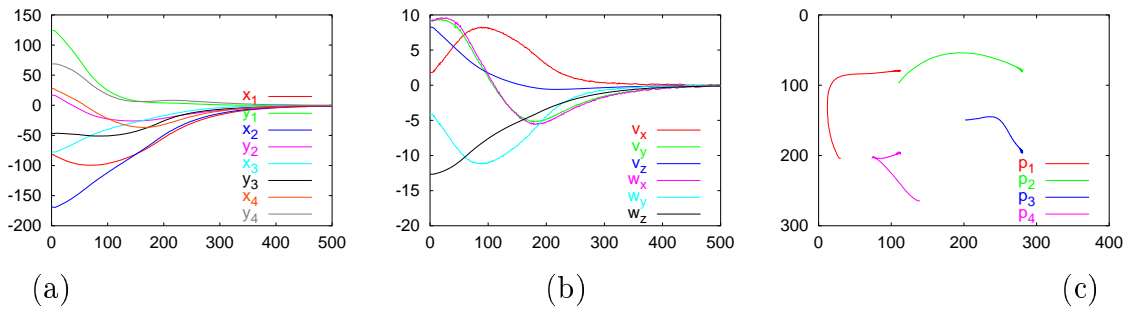


FIG. 5.27 – Résultats obtenus en utilisant les coordonnées de points comme informations visuelles : (a) erreurs sur les coordonnées de points (pixels), (b) vitesses de la caméra, (c) Trajectoires des points dans l'image

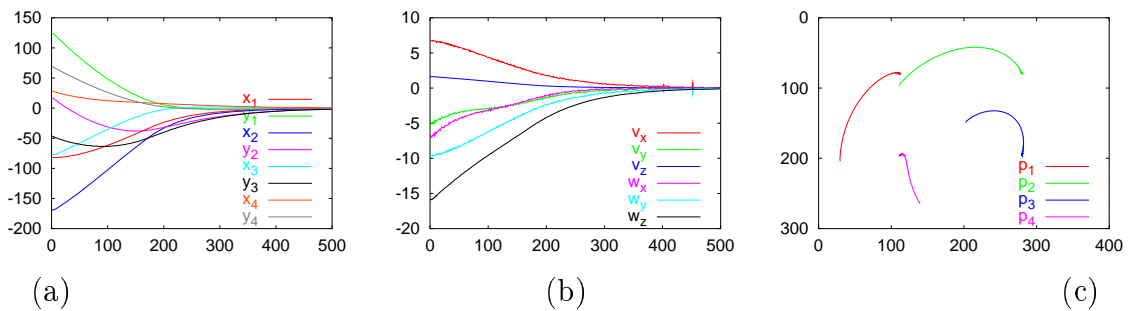


FIG. 5.28 – Résultats obtenus en utilisant les moments comme informations visuelles : (a) Erreurs sur les coordonnées de points (pixels), (b) vitesses de la caméra, (c) Trajectoire des points dans l'image

5.2.5 Résultats expérimentaux avec les paramètres de la pose

Dans ce paragraphe, nous comparons le comportement du système en utilisant les différentes valeurs suivantes des matrices d'interaction dans la boucle de commande :

- \mathbf{L}_s est la valeur courante de la matrice d'interaction calculée avec les paramètres courants du plan objet dans le repère caméra.
- \mathbf{L}_{s^*} est la valeur désirée de la matrice d'interaction.
- $\frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$ est la moyenne des valeurs des matrices d'interaction calculées pour les positions désirée et courante en injectant les paramètres estimés du plan objet dans le repère caméra.
- \mathbf{L}_s^{\parallel} est la valeur courante de la matrice d'interaction calculée avec les paramètres du plan estimés pour une position parallèle au plan image.
- $\frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel})$ est la moyenne des valeurs des matrices d'interaction calculées pour les positions désirée et courante calculées avec les paramètres du plan calculés pour une position parallèle au plan image.

Nous avons considéré deux positions initiales de la caméra référencées respectivement par les images données sur les figures 5.29.b et 5.29.c. La position désirée de la caméra est telle que les plans de l'image et de l'objet soient parallèles. Dans le deuxième cas, la variation de profondeur considérée est plus importante que dans le premier pour tester l'effet de la non linéarité de la matrice.

Cas 1 :

Les valeurs des informations visuelles et les vitesses de la caméra obtenues pour cette expérience sont données respectivement sur les figures 5.30 et 5.31. A partir de ces deux figures, on peut noter que :

- dans le cas où $\widehat{\mathbf{L}}_s = \mathbf{L}_s$, les erreurs sur les informations visuelles décroissent exponentiellement. Cependant, on observe sur la figure 5.31.a une petite oscillation sur une des composantes de la vitesse de la caméra.
- dans le cas où $\widehat{\mathbf{L}}_s = \mathbf{L}_{s^*}$ le comportement des vitesses, ainsi que celui des informations visuelles est satisfaisant, excepté une petite oscillation (courbe en vert de la figure 5.30.b).
- dans le cas de $\widehat{\mathbf{L}}_s = \frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$, les résultats sont très satisfaisants (voir 5.30.c et 5.31.c)
- dans le cas où $\widehat{\mathbf{L}}_s = \mathbf{L}_s^{\parallel}$, le comportement du système est moins intéressant que dans les cas précédents. En effet, on obtient une décroissance brutale de certaines composantes de la vitesse de la caméra, alors que d'autres décroissent lentement. On peut noter également que le comportement des vitesses et des informations visuelles ne se fait pas de la même manière.
- $\frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel})$, le comportement du système est satisfaisant, étant donné que les paramètres de la pose ne sont pas estimés pour la position courante.

Les résultats obtenus sont globalement satisfaisants excepté le cas où $\widehat{\mathbf{L}}_{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}^{\parallel}$. Néanmoins, le comportement du système en utilisant la matrice $\frac{1}{2}(\mathbf{L}_{\mathbf{s}^*} + \mathbf{L}_{\mathbf{s}})$ semble être le meilleur.

Cas 2 :

Dans cette seconde expérience qui nécessite un déplacement plus important, nous avons considéré les mêmes choix de la matrice d'interaction que dans l'expérience précédente. Les résultats sont donnés sur les figures 5.32 et 5.33. Les cas où $\widehat{\mathbf{L}}_{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}^{\parallel}$ et $\widehat{\mathbf{L}}_{\mathbf{s}} = \frac{1}{2}(\mathbf{L}_{\mathbf{s}}^{\parallel} + \mathbf{L}_{\mathbf{s}^*}^{\parallel})$ ne sont pas donnés sur ces figures, étant donné que, dans le premier cas, le système a divergé et, dans le deuxième cas, l'objet a quitté le champ de vision de la caméra pendant la phase de convergence. À partir de ces figures, on peut noter de nouveau le bon comportement des informations visuelles et des composantes de la vitesse de la caméra.

Finalement, la figure 5.34 donne les trajectoires de la caméra obtenues en utilisant chacune des matrices d'interaction considérées pour les deux positions initiales de la caméra. Cette figure montre que les trajectoires obtenues sont globalement satisfaisantes. Cela confirme que la matrice d'interaction associée à notre choix d'informations visuelles varie peu entre \mathbf{s} et \mathbf{s}^* , ce qui était l'objectif majeur de notre travail.

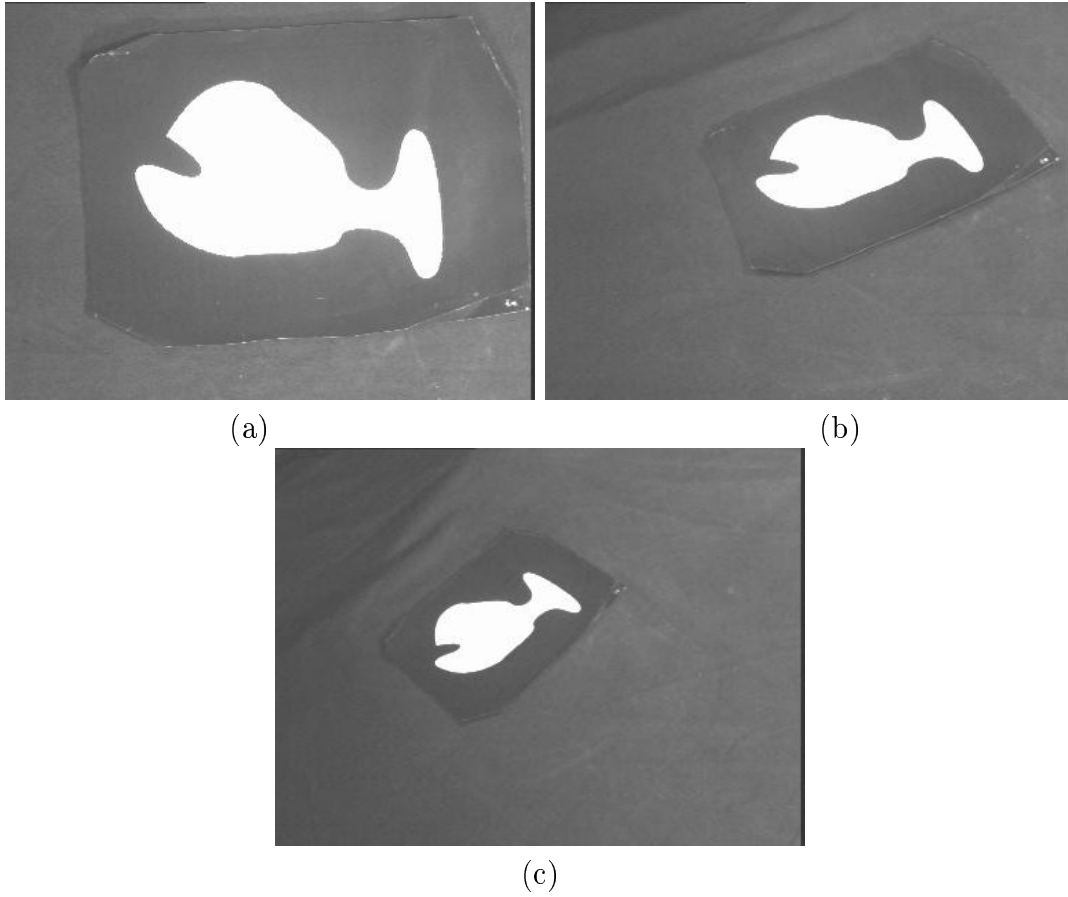


FIG. 5.29 – Images utilisées : (a) image désirée, (b) image initiale utilisée pour l'expérience 1, (c) image initiale utilisée pour l'expérience 2.

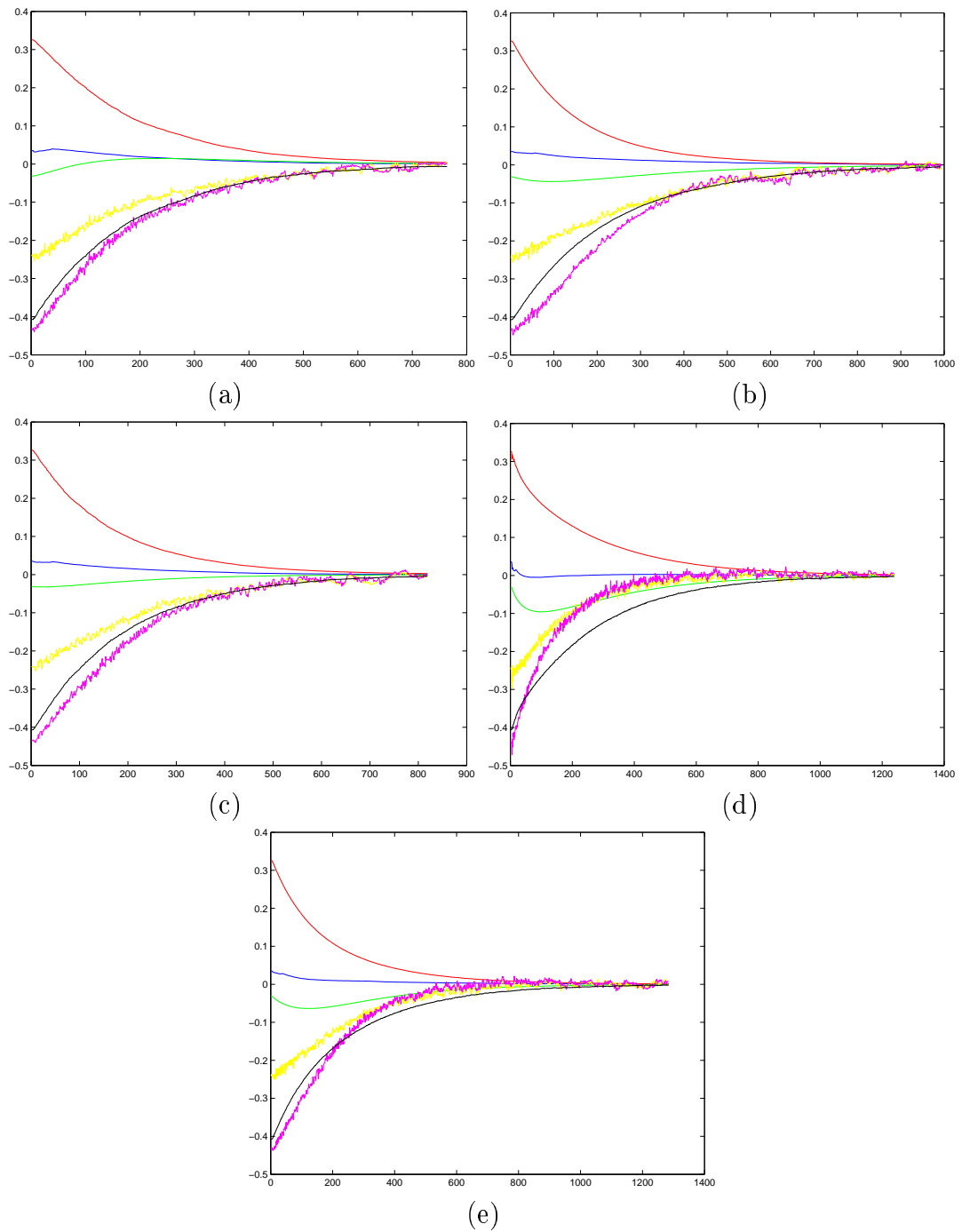


FIG. 5.30 – Erreurs sur les informations visuelles selon la valeur de la matrice d'interaction obtenues pour l'expérience 1: (a) résultat obtenu en utilisant \mathbf{L}_s , (b) résultat obtenu en utilisant \mathbf{L}_{s^*} , (c) résultat obtenu en utilisant $\frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$, (d) résultat obtenu en utilisant \mathbf{L}_s^{\parallel} , (e) résultat obtenu en utilisant $\frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel})$.

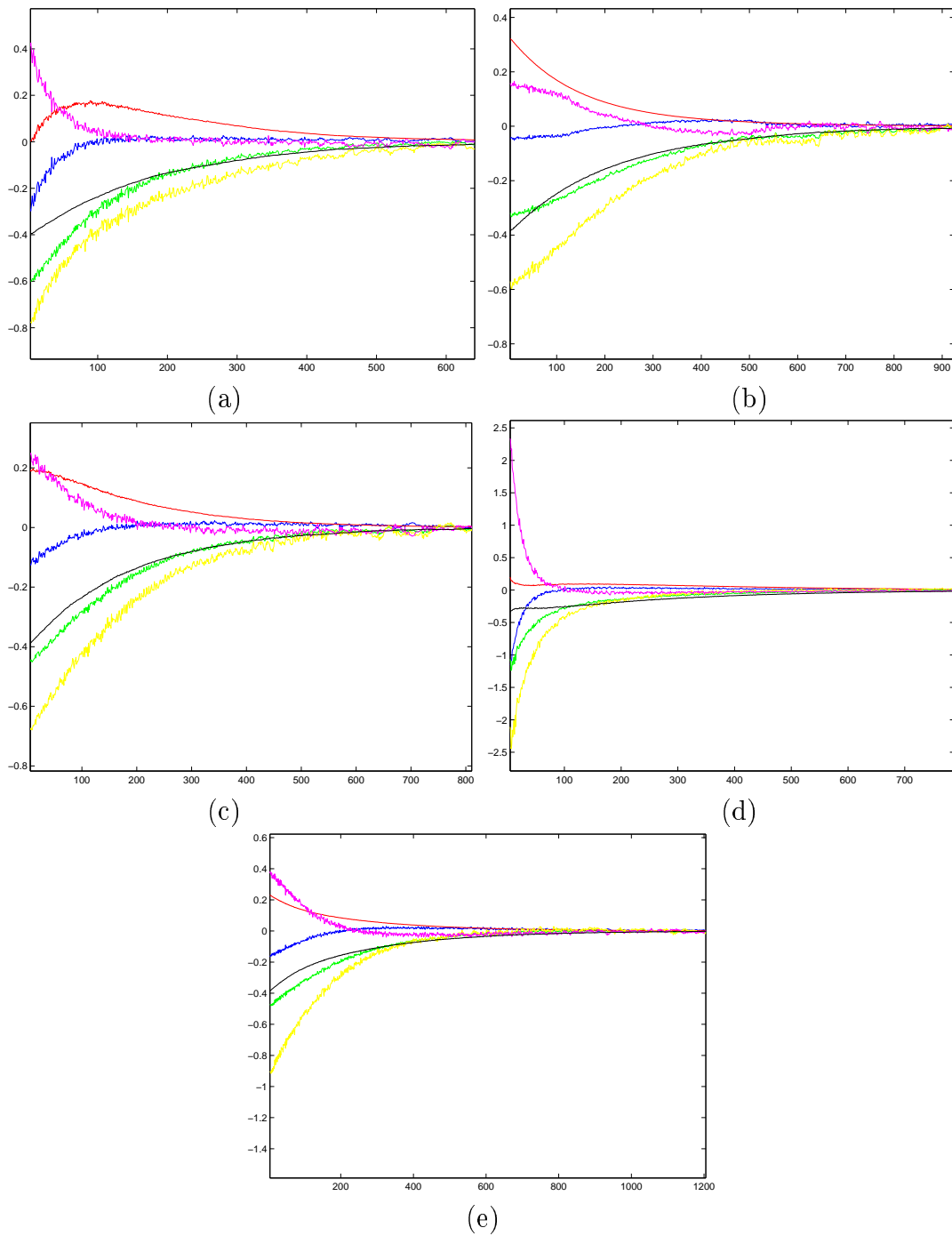


FIG. 5.31 – Vitesses de la caméra selon la valeur de la matrice d'interaction obtenues pour l'expérience 1: (a) résultat obtenu en utilisant \mathbf{L}_s , (b) résultat obtenu en utilisant \mathbf{L}_{s^*} , (c) résultat obtenu en utilisant $\frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$, (d) résultat obtenu en utilisant \mathbf{L}_s^{\parallel} , (e) résultat obtenu en utilisant $\frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel})$.

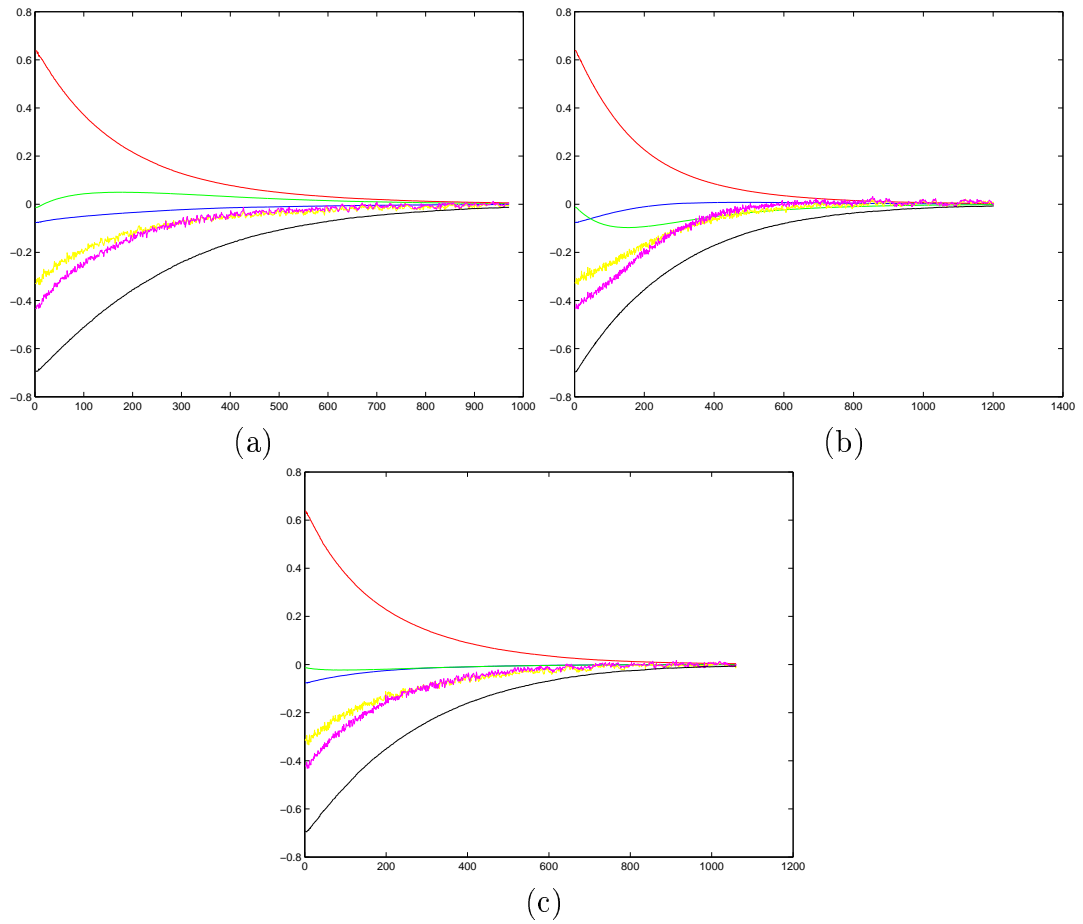


FIG. 5.32 – Erreurs sur les informations visuelles selon la valeur de la matrice d'interaction obtenues pour l'expérience 2: (a) résultat obtenu en utilisant \mathbf{L}_s , (b) résultat obtenu en utilisant \mathbf{L}_{s^*} , (c) résultat obtenu en utilisant $\frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$.

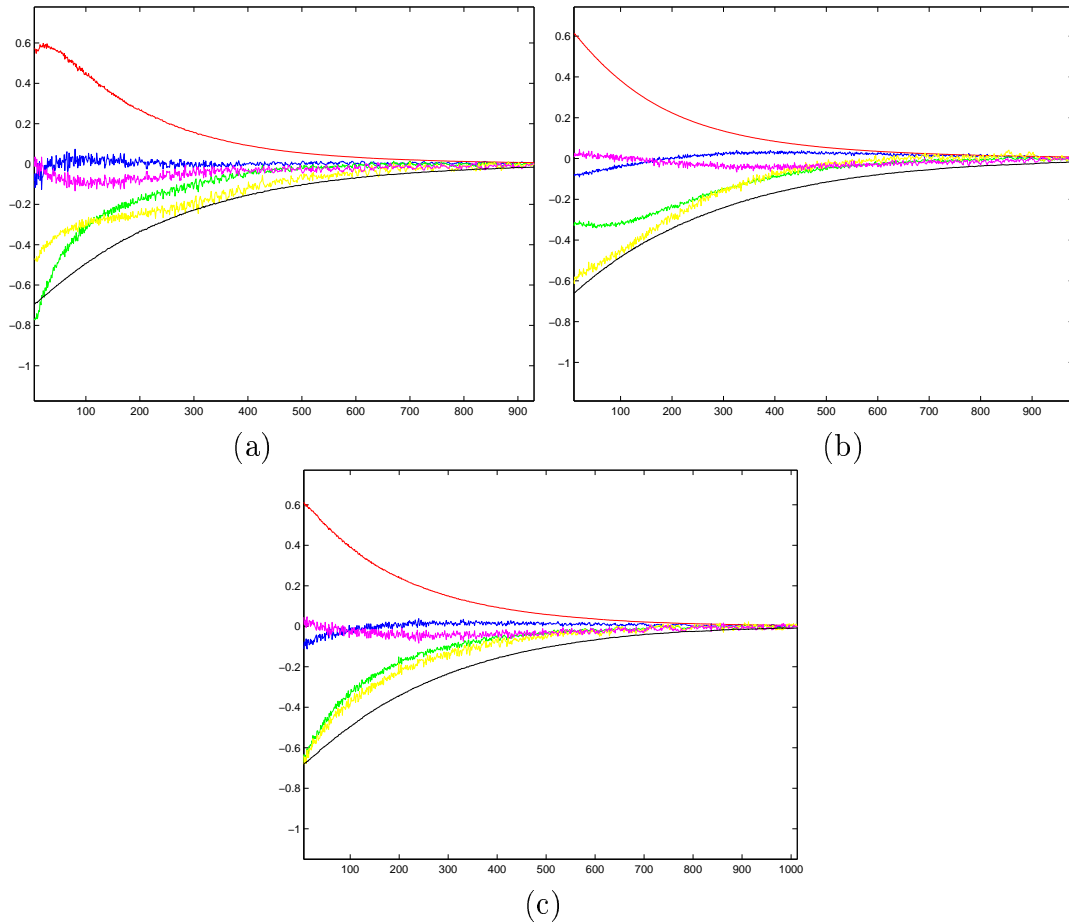
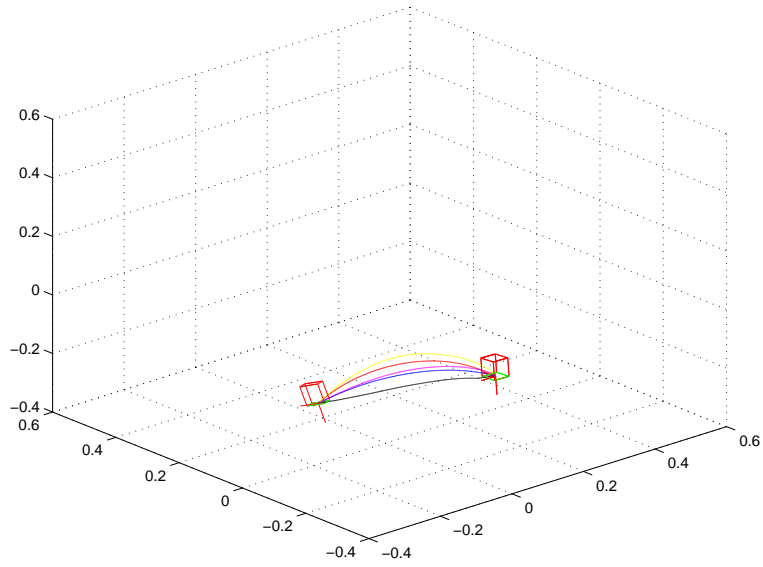
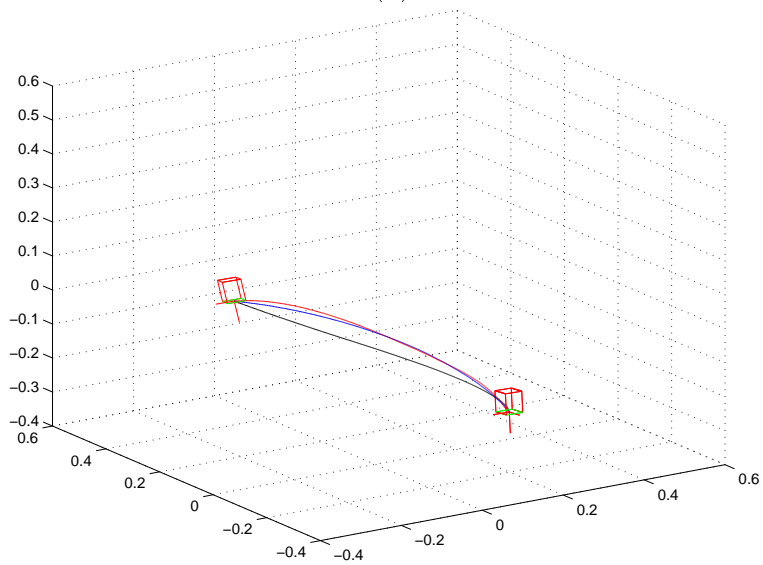


FIG. 5.33 – Vitesses de la caméra selon la valeur de la matrice d'interaction obtenues pour l'expérience 2: (a) résultat obtenu en utilisant \mathbf{L}_s , (b) résultat obtenu en utilisant \mathbf{L}_{s^*} , (c) résultat obtenu en utilisant $\frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$.



(a)



(b)

FIG. 5.34 – Trajectoire de la caméra selon le choix de la matrice d'interaction : (a) résultat obtenu pour l'expérience 1, (b) résultat obtenu pour l'expérience 2. (en rouge) résultat obtenu en utilisant \mathbf{L}_s , (en noir) résultat obtenu en utilisant \mathbf{L}_{s^*} , (en bleu) résultat obtenu en utilisant $\frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$, (en jaune) résultat obtenu en utilisant \mathbf{L}_s^{\parallel} , (en violet) résultat obtenu en utilisant $\frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel})$.

5.3 Résultats expérimentaux : cas d'un objet discret

Dans cette section, nous présentons les résultats expérimentaux obtenus en utilisant un objet discret. Deux cas de positions désirées sont une nouvelle fois traitées séparément : position du plan image parallèle à l'objet et position non parallèle. L'image correspondant à chaque position est donnée sur les figures 5.35.a et 5.35.b. Nous avons fixé approximativement la profondeur désirée Z^* à 0.5 m dans les deux cas. La rotation virtuelle \mathbf{R} a été spécifiée par une rotation de 30 dg autour de l'axe x de la caméra dans le cas non parallèle. Finalement, des résultats obtenus dans le cas d'une image complexe (ensemble de points extraits en utilisant un détecteur de Harris) seront également présentés.

5.3.1 Mouvement de translation pure

Nous avons considéré dans un premier temps le cas d'un mouvement de translation pure entre la position initiale et la position désirée de la caméra. Dans ces expériences, le même mouvement de translation $\mathbf{T} = [-24cm, 17cm, -70cm]$ a été considéré dans le cas parallèle comme dans le cas non parallèle (voir figure 5.35). Nous avons comparé les résultats obtenus à l'aide des combinaisons de moments que nous proposons (x_n, y_n, a_n) et les coordonnées de points (x_k, y_k) . Dans le cas où les coordonnées de points sont utilisées la décroissance exponentielle n'est pas obtenue (voir figure 5.36) et la trajectoire de la caméra n'est donc pas une ligne droite. Dans les deux cas de figure de position désirée de la caméra, on peut noter à partir de la figure 5.36 les larges améliorations obtenues en utilisant nos informations visuelles et la rotation virtuelle pour se ramener à une position parallèle. En effet, elles permettent d'obtenir une pure décroissance exponentielle des erreurs sur les informations visuelles comme sur les vitesses de la caméra. La trajectoire de la caméra est une ligne droite dans les deux cas de figure considérés.

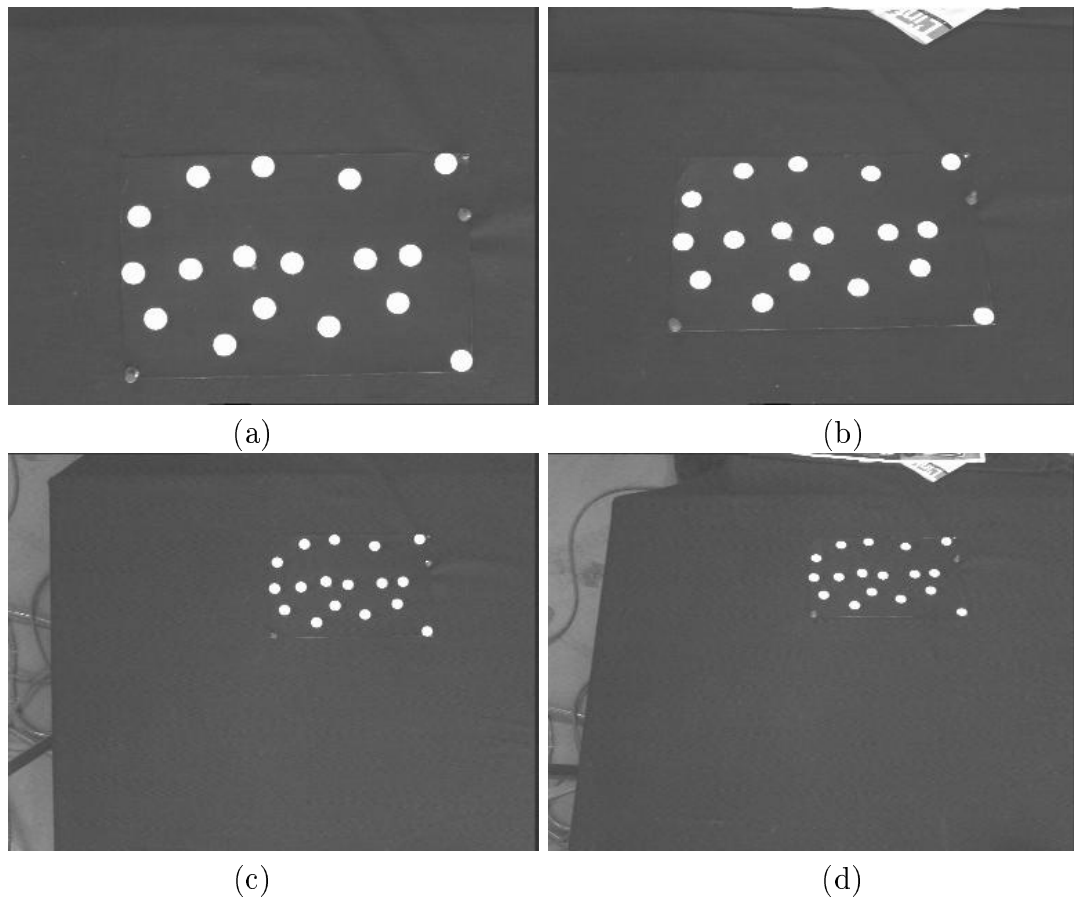


FIG. 5.35 – (a) image désirée où l'objet est parallèle au plan image, (b) image désirée où l'objet est non parallèle au plan image, (c) image initiale pour le mouvement de translation pure entre (a) and (c), (d) image initiale pour le mouvement de translation pure entre (b) and (d).

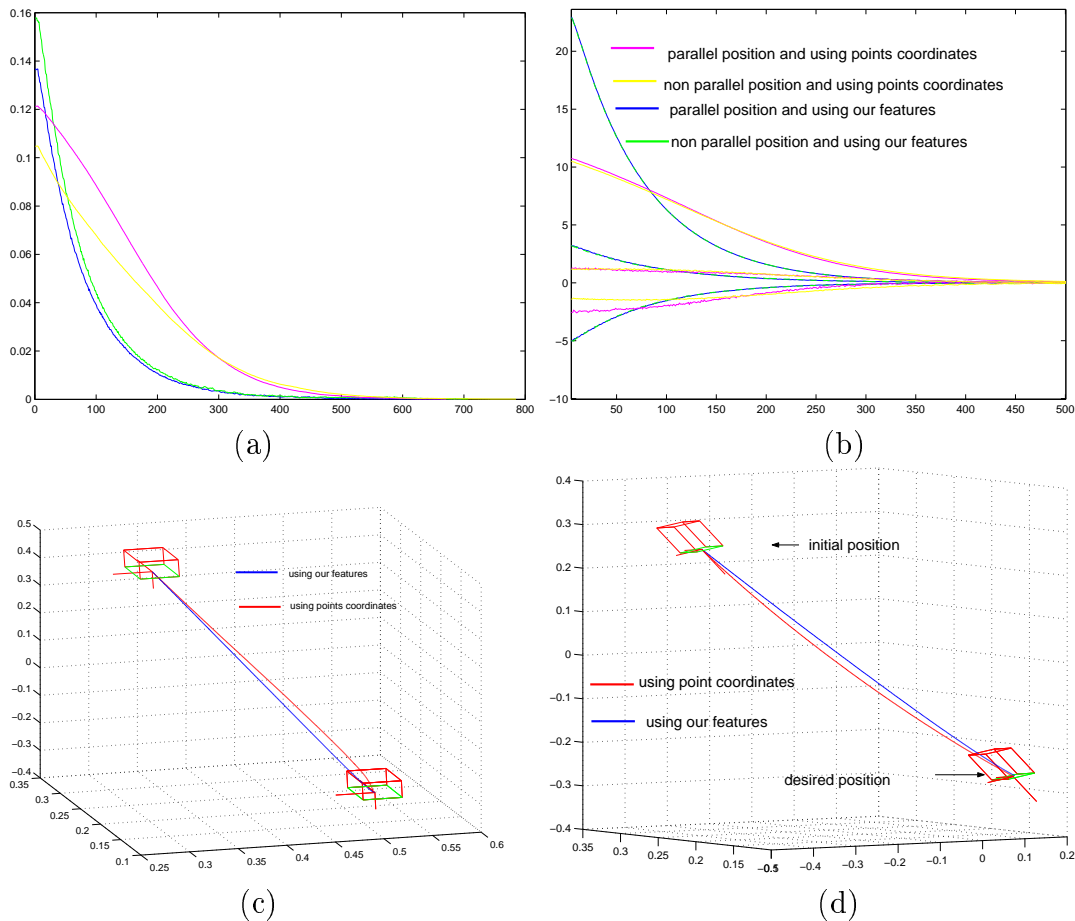


FIG. 5.36 – Résultats obtenus pour un mouvement de translation pure : (a) moyenne des erreurs sur les informations visuelles, (b) vitesses de la caméra, (c) trajectoire 3D de la caméra dans le cas où les plan image et objet sont parallèles, (d) trajectoire 3D de la caméra dans cas où les plans image et objet ne sont non parallèles.

5.3.2 Mouvement complexe

Nous considérons maintenant le cas d'un mouvement complexe mêlant un mouvement de translation et un mouvement de rotation pour une position désirée non parallèle au plan objet. Plus précisément, le déplacement à réaliser est approximativement une translation de 65 cm et une rotation de 30 dg. Les images initiale et désirée sont données par la figure 5.37.a and 5.35.b. Pour cette position et après la rotation virtuelle de 30° autour de l'axe x , on a :

$$\mathbf{L}_{\mathbf{s}(s_i^*)}^{\parallel} = \begin{pmatrix} -1 & 0 & 0 & 0.00 & -0.50 & 0.00 \\ 0 & -1 & 0 & 0.50 & -0.0 & 0.01 \\ 0 & 0 & -1 & -0.00 & -0.01 & 0 \\ 0 & 0 & 0 & 1.49 & -0.56 & 0 \\ 0 & 0 & 0 & 1.58 & 2.50 & 0 \\ 0 & 0 & 0 & 0.11 & -0.17 & -1 \end{pmatrix} \quad (5.8)$$

Nous avons choisi pour cette série d'expériences le couple (r_9, r_{10}) pour contrôler les deux vitesses de rotation ω_x et ω_y . Comme dans le cas continu, la matrice d'interaction obtenue est creuse. Son conditionnement est égal à 3.34 (il vaut 95.61 en utilisant les coordonnées de points). Les résultats obtenus sont donnés par la figure 5.37. On note à partir de cette figure le bon comportement des informations visuelles et des vitesses: on obtient une décroissance exponentielle des vitesses et des erreurs sur les informations visuelles sans oscillation (voir figure 5.37.b et figure 5.37.d). La trajectoire de la caméra est également satisfaisante (voir figure 5.37.c).

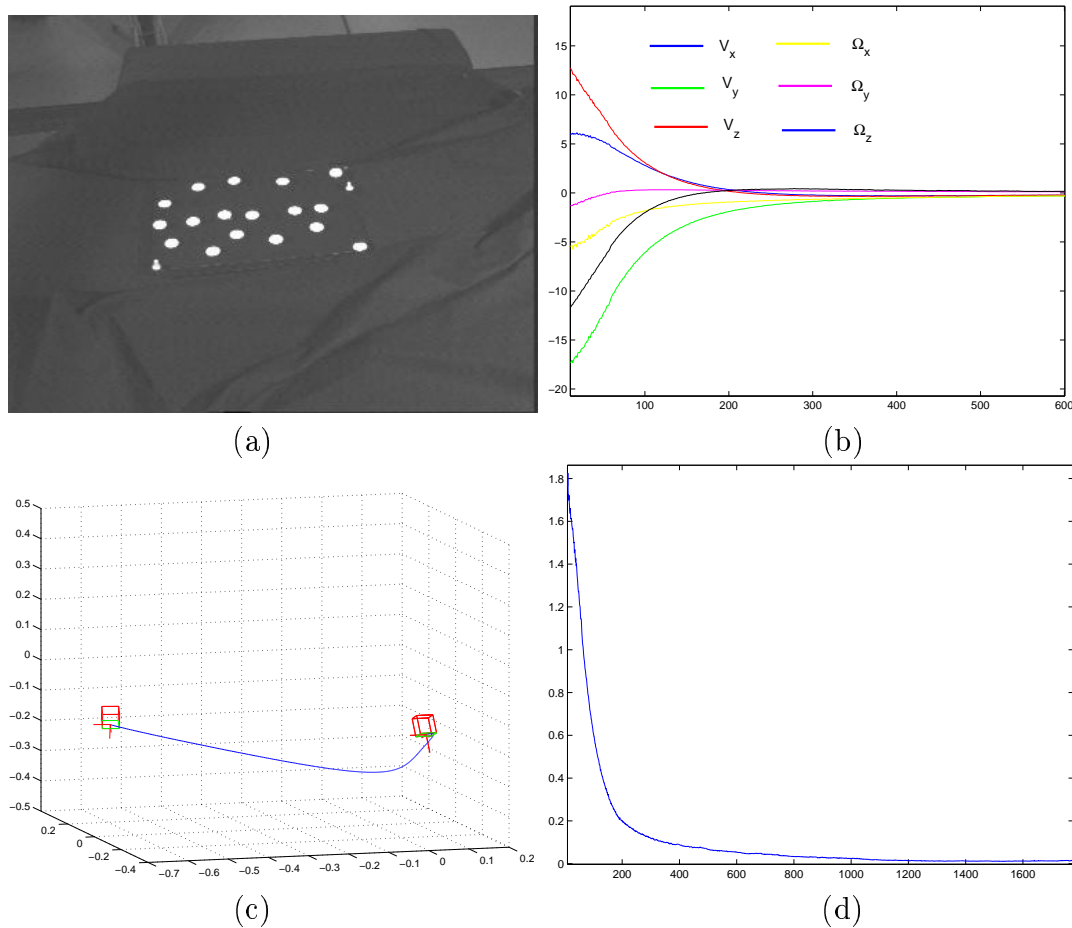


FIG. 5.37 – Résultats obtenus pour un mouvement complexe: (a) image initiale, (b) vitesses de la caméra, (c) trajectoire de la caméra, (d) erreur moyenne sur les informations visuelles

5.3.3 Résultats avec des erreurs de modélisation

Nous testons maintenant la robustesse de notre méthode vis-à-vis d'erreurs de modélisation. Dans cette expérience, des erreurs ont été introduites sur les paramètres intrinsèques de la caméra (25% sur la longueur de la focale et 20 pixels sur les coordonnées du point principal) et sur la profondeur de l'objet par rapport à la caméra dans sa position désirée. Une erreur égale à 10 dg sur la rotation \mathbf{R} a été également introduite. Les résultats obtenus sont donnés sur la figure 5.38. Le comportement du système est similaire à celui obtenu dans le cas sans erreur de modélisation. Cela valide la robustesse de notre schéma de commande vis-à-vis de ces erreurs.

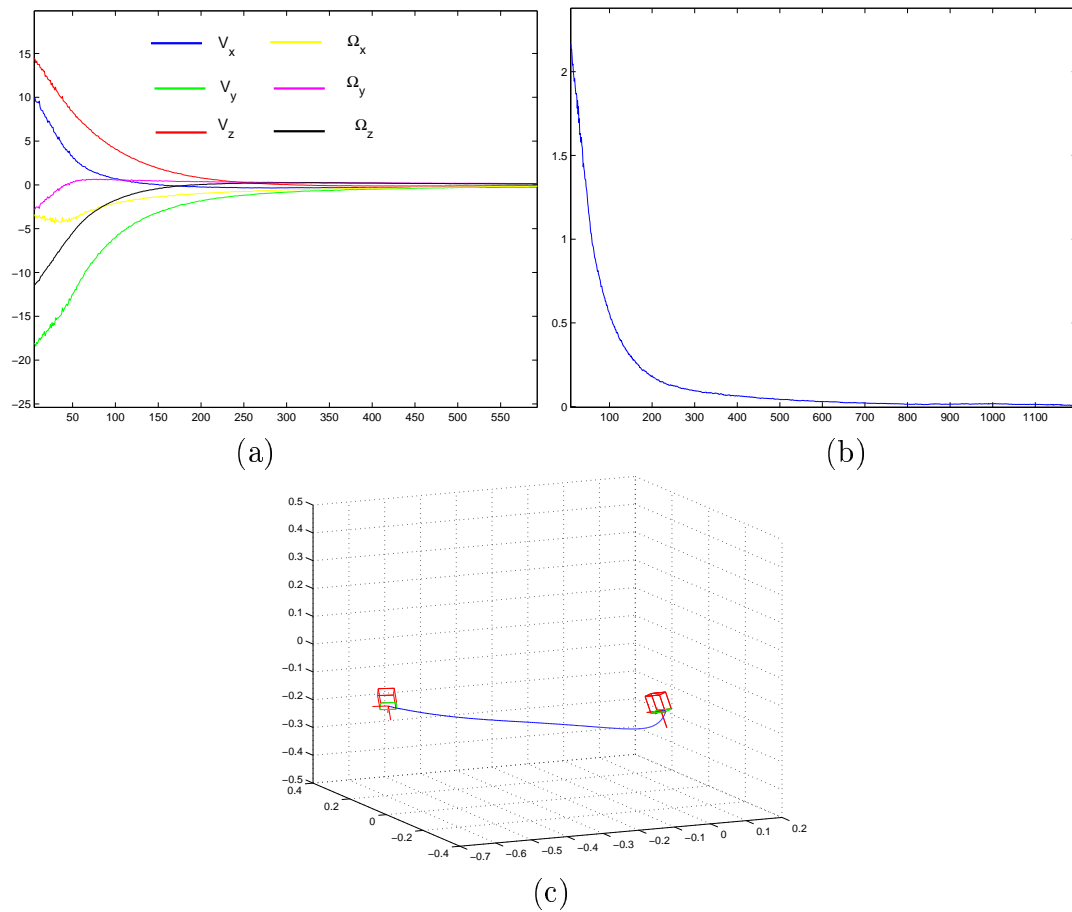


FIG. 5.38 – Résultats avec des erreurs de modélisation: (a) vitesses de la caméra, (b) erreur moyenne sur les informations visuelles, (c) trajectoire de la caméra.

5.3.4 Résultats en utilisant des images complexes

Dans ce paragraphe, nous présentons des résultats similaires mais obtenus avec des images plus complexes (voir figure 5.39). Les points considérés ont été extraits en utilisant un détecteur de Harris et suivis à l'aide de l'algorithme SSD. On peut noter toutefois que les courbes obtenues sont plus bruitées que dans le cas où de simples points sont utilisés. Les composantes les plus bruitées sont celles de ω_x et ω_y , ce qui est logique puisque que les valeurs des informations visuelles correspondantes dépendent des moments d'ordre 5 (alors que ω_z et v_z ne sont pas sensibles puisqu'ils dépendent seulement des moments d'ordre 2). Néanmoins, la décroissance exponentielle, la convergence et la stabilité sont obtenues, ce qui valide notre approche. Ces résultats pourraient encore être améliorés en utilisant un algorithme de suivi à précision sub-pixelique comme par exemple celui de Shi-Tomasi [Shi 94].

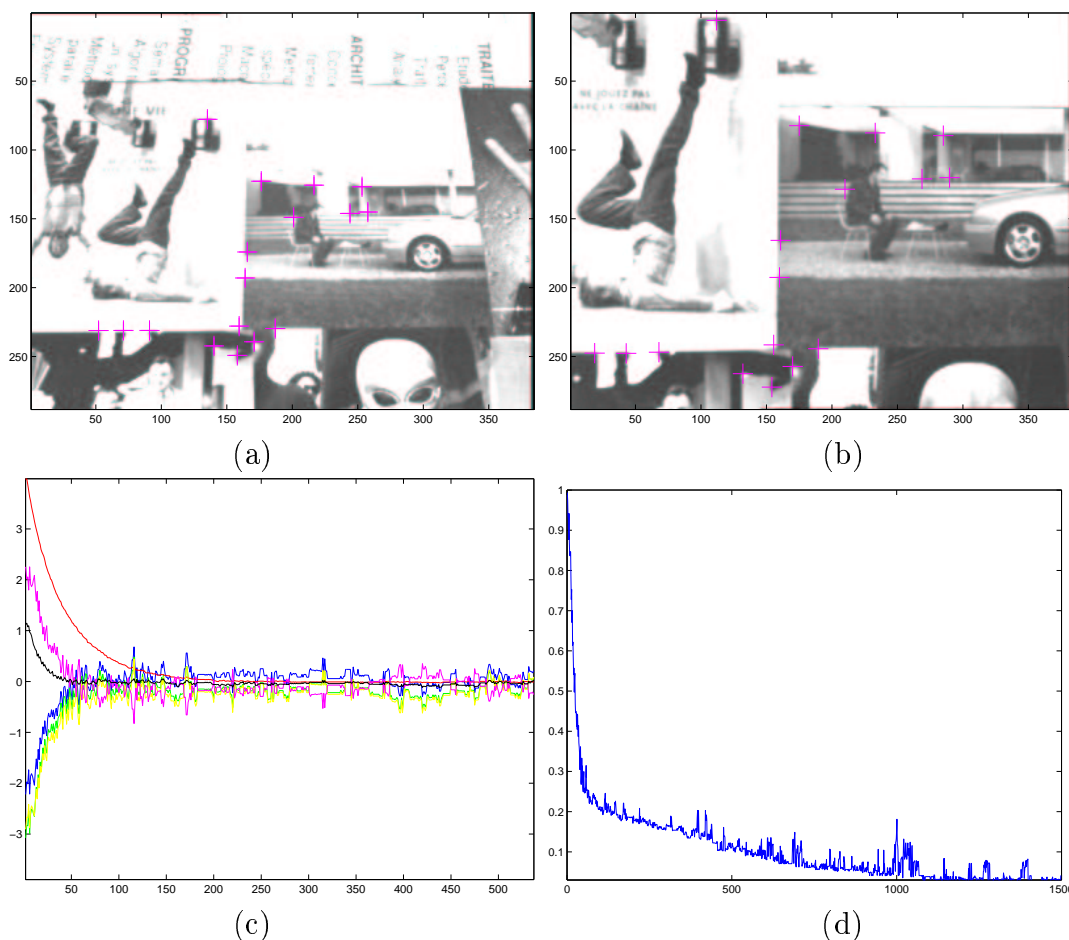


FIG. 5.39 – Résultats en utilisant des images complexes: (a) image initiale, (b) image désirée, (c) vitesses de la caméra, (d) erreur moyenne sur les informations visuelles

5.4 Conclusion

Dans ce chapitre, nous avons validé les résultats théoriques sur le calcul des invariants et le choix des informations visuelles obtenus respectivement dans les chapitres 2 et 3. Les expériences ont été réalisées en utilisant une caméra montée sur l'organe terminal d'un robot à six degrés de liberté.

Dans un premier temps, nous avons discuté du choix des deux invariants r_i et r_j à utiliser dans la boucle de commande pour plusieurs objets: une "baleine", un "cerveau" et un ensemble de points. Le choix de ces deux informations visuelles s'est basé sur leur comportement vis-à-vis des mouvements de rotation selon l'axe x et selon l'axe y de la caméra. Après avoir choisi les couples d'invariants pour chaque objet considéré, nous avons présenté une première série d'expériences en utilisant la valeur moyenne des matrices d'interaction calculées pour les positions désirée et courante sans calcul des paramètres du plan de l'objet dans le repère caméra. L'ensemble de ces expériences montre que le système converge pour de très importants déplacements. Nous avons également testé la robustesse aux erreurs de modélisation et occultation initiale d'objet. Dans la deuxième série d'expérience, nous avons comparé l'utilisation des différentes valeurs \mathbf{L}_s , \mathbf{L}_{s^*} , $\frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$, \mathbf{L}_s^{\parallel} et $\frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel})$. Les résultats obtenus sont satisfaisants en utilisant les différentes matrices précédentes, sauf dans le cas où la matrice \mathbf{L}_s^{\parallel} .

Chapitre 6

Conclusions et perspectives

6.1 Conclusion

Le travail que nous avons présenté dans ce mémoire s'est attaché à résoudre un ensemble de problèmes potentiels en asservissement visuel et en calcul de pose. En effet, nous avons étudié la détermination d'une représentation optimale d'un objet dans l'image pour résoudre les problèmes suivants : trouver une représentation pour traiter des objets de formes complexes, réaliser un schéma de commande découplé et finalement, minimiser les non linéarités du système. Pour ce faire, nous avons procédé en quatre parties, que nous rappelons ici :

- calcul des moments invariants;
- choix des informations visuelles;
- validation en calcul de pose;
- validation en asservissement visuel.

Calcul des moments invariants

Dans un premier temps, nous avons décrit les principales techniques de calcul de moments invariants. Plus précisément nous avons rappelé quelques notions sur la théorie des invariants algébriques ainsi que le théorème fondamental des moments invariants, qui sont la base de nombreux travaux sur le sujet. Nous avons alors proposé une nouvelle méthode basée sur un calcul variationnel pour déterminer des formules analytiques de moments invariants. La méthode présentée a permis de déterminer des moments invariants aux rotations, aux translations et aux changements d'échelle. Elle est valable dans un espace de dimension quelconque et elle permet une détermination systématique des invariants aux transformations mentionnées précédemment.

Nous avons établi ensuite une règle simple pour construire une base de moments invariants et éliminer ainsi la redondance entre invariants. Cette règle est nécessaire pour éviter toute redondance inutile dans les informations visuelles utilisées par

la suite dans la loi de commande. Les résultats obtenus concernent aussi bien les moments calculés à partir d'une distribution continue que les moments calculés à partir d'un ensemble de points. La liste exhaustive de moments invariants a été déterminée dans le cas d'un espace de dimension 2 et d'un espace de dimension 3 et pour les moments d'ordre inférieur à 6.

Choix des informations visuelles

Dans cette partie, nous avons rappelé les problèmes potentiels en asservissement visuel : difficulté de représenter des objets de formes complexes, couplage de la commande et non linéarité de la matrice d'interaction. Pour remédier à ces problèmes, nous avons proposé une méthode d'asservissement visuel basée sur des combinaisons de moments. Ainsi, nous avons calculé la matrice d'interaction associée aux moments calculés à partir d'une distribution continue et à partir d'un ensemble discret de points. Les résultats obtenus dans les deux cas sont similaires, mais présentent tout de même de légères différences.

Nous avons traité également deux sortes de projection dans l'image : tout d'abord la projection perspective classique sur un plan, puis la projection sur une sphère. Pour résoudre les problèmes du couplage de la commande et de la non linéarité de la commande, nous avons exploité les résultats obtenus dans la première partie. En effet, dans un premier temps, nous avons étudié l'utilisation des invariants aux rotations calculés à partir d'une projection sphérique pour contrôler les mouvements de translation du robot. Il s'est avéré à l'issue de cette étude que ces invariants dépendent principalement du mouvement de translation selon l'axe optique de la caméra. Cela rend impossible leur utilisation sous leurs formes actuelles pour contrôler les trois degrés de liberté de translations. Nous avons alors proposé des informations visuelles basées sur les invariants d'une projection perspective plane pour obtenir une matrice triangulaire lorsque l'objet est parallèle au plan image. En outre, pour ces positions, les éléments de la matrice d'interaction qui correspondent aux translations forment un bloc diagonal constant. Ces résultats ont été généralisés au cas où les plans de l'image et celui de l'objet dans sa position désirée ne sont pas parallèles.

Validation en calcul de pose

Nous avons traité le problème du calcul de pose comme étant un problème dual de l'asservissement visuel. En effet, le problème du calcul de pose peut être considéré comme un problème de positionnement d'une caméra virtuelle en utilisant un asservissement 2D (AVV). Nous avons alors présenté une nouvelle méthode basée sur ce principe et en utilisant les bonnes propriétés de découplage obtenues dans la partie consacrée au choix des primitives. Pour accélérer le calcul, nous avons proposé une méthode basée sur un développement de Taylor permettant d'approximer les valeurs des moments après chaque mouvement de rotation. Puis, nous avons validé

l'ensemble de ces résultats dans le cas continu et dans le cas discret. Cela a montré que la zone de convergence est très large et qu'une initialisation grossière suffit pour cet algorithme. Cependant, comme pour toute méthode itérative, quand le mouvement de rotation à estimer est très important, notre approche peut converger vers un minimum local. Pour remédier à ce problème, nous avons proposé une nouvelle méthode utilisant un algorithme d'optimisation exhaustif. elle permet de fournir une bonne approximation du minimum global, qui pourra par la suite servir comme initialisation à la méthode utilisant l'AVV. L'utilisation des moments invariants dans cette méthode a permis de limiter la recherche exhaustive sur les deux degrés de liberté de rotation autour des axes x et y de la caméra. Les mouvements selon les autres degrés de liberté sont obtenus par des formules analytiques.

Les deux méthodes de calcul de pose proposées dans ce travail estiment dans une première étape le mouvement de rotation. En pratique, cette étape ne nécessite pas la connaissance du modèle de l'objet. En effet, l'estimation du mouvement de rotation de la pose suppose juste la connaissance des invariants r_i , r_j et θ pour une position de la caméra où les plans image et objet sont parallèles. pour cela, nous avons développé une nouvelle approche basée sur un algorithme de calcul exhaustif similaire à celui utilisé dans la deuxième méthode de calcul de pose. En effet, notre approche permet d'estimer le mouvement de rotation entre deux positions de la caméra, la translation à un facteur d'échelle près, ainsi que l'orientation du plan objet dans un des repère lié à la caméra. La détermination de cette orientation nous permet de calculer les r_i , r_j et θ pour une position de la caméra où les plans image et objet sont parallèles. Nous avons alors adapté les méthodes de calcul de pose proposées dans ce chapitre pour le cas où l'on ne dispose pas du modèle de l'objet.

Validation en asservissement visuel

Après le calcul de pose, nous avons validé les résultats théoriques sur le calcul des invariants et le choix des informations visuelles par des expériences réalisées en utilisant un système robotique à six degrés de liberté. Nous avons utilisé des objets continus et des objets discrets de formes asymétriques dans les séries d'expériences présentées. Dans un premier temps, nous avons présenté une série d'expériences en utilisant $\frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel})$ comme approximation de la matrice d'interaction. Les résultats obtenus montrent que le système converge pour d'importants déplacements à réaliser et sans estimation en ligne des paramètres courants du plan objet dans le repère caméra. Dans la deuxième série d'expériences, nous avons comparé l'utilisation des différentes valeurs \mathbf{L}_s , \mathbf{L}_{s^*} , $\frac{1}{2}(\mathbf{L}_{s^*} + \mathbf{L}_s)$, \mathbf{L}_s^{\parallel} et $\frac{1}{2}(\mathbf{L}_s^{\parallel} + \mathbf{L}_{s^*}^{\parallel})$. Les résultats en utilisant ces différentes matrices sont en général similaires et satisfaisants (excepté pour \mathbf{L}_s^{\parallel}), ce qui valide notre recherche d'obtention d'une matrice variant peu dans l'espace de travail.

Dans la section suivante, nous présentons des perspectives pour améliorer les résultats obtenus dans cette thèse.

6.2 Perspectives

Pour améliorer et étendre les résultats obtenus dans ce travail, nous pouvons envisager de développer les pistes suivantes :

- chercher des invariants permettant un découplage global;
- travailler sur l'appariement de courbes fermées dans une image pour traiter des scènes plus complexes;
- appliquer des techniques d'asservissement robuste pour remédier au problème de données aberrantes;
- coupler asservissement visuel et planification de trajectoires dans l'image pour résoudre les problèmes de convergence en général (visibilité, évitement d'occultation par exemple).

De nouveaux invariants

Dans le chapitre 3, nous avons proposé d'utiliser un couple d'invariants pour contrôler les mouvements de rotations. En fait, l'invariance de ces informations visuelles est valable dans le cas où les plans objet et image sont parallèles. Néanmoins, certaines parmi ces informations visuelles sont invariantes aux mouvements de translation selon l'axe x et selon l'axe y quelle que soit l'orientation du plan objet par rapport au repère caméra. En effet, le rapport suivant :

$$I_{t_1} = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{m_{00}^4}$$

est invariant à v_x et à v_y quelle que soit les paramètres du plan objet dans le repère caméra. Cependant, les valeurs numériques des ces invariants sont en général très faibles. Cela pose des problèmes pour leur utilisation. Pour résoudre ce problème, on peut espérer trouver d'autres invariants en utilisant d'autres normalisations, permettant d'obtenir des résultats numériquement acceptables.

Nous avons également vu au chapitre 3 qu'il est possible d'obtenir des invariants 3D aux mouvements de rotation issus d'une projection sphérique. L'utilisation de ces invariants pose également des problèmes numériques, puisqu'ils dépendent principalement du mouvement de translation selon l'axe optique de la caméra. Il serait intéressant de trouver des transformations à appliquer sur ces invariants pour obtenir d'autres invariants avec des dynamiques similaires sur les 3 axes de translation. Par ailleurs, il serait intéressant de voir si ces problèmes numériques se rencontrent encore dans le cas d'un capteur de vision omnidirectionnelle.

Enfin, le couple d'invariants utilisé pour contrôler les mouvements de rotation selon les axes x et y dépend pour le moment de la valeur des moments, et donc de l'objet considéré. Il serait également important de trouver un couple d'invariants unique pour tout type d'objets.

Appariement de courbes fermées dans une image

Dans le cas d'objets continus, nous avons supposé qu'un seul objet est présent dans le champ de vision de la caméra. Pour généraliser notre approche à des scènes très complexes, où de nombreux objets peuvent être inclus dans le champ de vision de la caméra, il est nécessaire d'éliminer les objets qui ne sont pas en commun entre l'image courante et l'image désirée.

Dans le cas où les objets inclus dans l'image désirée peuvent être décrits par des courbes fermées, il existe déjà des techniques d'appariement utilisant généralement des invariants aux transformations affines (où encore aux rotations, aux translations et aux changements d'échelle) pour comparer la similitude de formes données [Wong 78]. Cependant, comme nous l'avons mentionné au chapitre 4, la transformation affine n'est valable que si l'on se restreint à un modèle orthographique de la caméra. Cela pose un problème pour distinguer des formes similaires pouvant être confondues avec la courbe à appairer.

Pour résoudre ce problème, nous proposons d'exploiter les résultats obtenus dans la partie calcul de pose. En effet, à l'aide de ces techniques, il est possible de déterminer une "pose" qui minimise les erreurs sur les informations visuelles calculées à partir de la courbe extraite dans l'image désirée et une courbe extraite dans l'image courante. Une fois la pose estimée, il suffit de comparer les erreurs sur les moments calculés sur les deux courbes pour trouver la courbe correspondante.

Asservissement robuste aux bruits de mesure et aux problèmes de données aberrantes

Dans ce manuscrit, outre les objets continus, nous avons également appliqué notre approche au cas où l'objet est constitué d'un ensemble de points. Dans ce cas, nous avons supposé que l'ensemble des points extraits de l'image désirée et celui extrait de l'image courante sont identiques pour réaliser la tâche de commande. Cette contrainte est plus souple qu'un appariement point à point nécessaire dans le cas où l'asservissement visuel est réalisé en utilisant les coordonnées de points comme informations visuelles. Pour préserver cet avantage, nous devons remédier aux éventuels problèmes de points aberrants (points extraits de l'image courante sans appartenir à l'image désirée ou vice-versa). Récemment, une nouvelle approche basée sur les M-estimateurs a été proposée pour résoudre ce problème dans le cas où l'asservissement est réalisé en utilisant les coordonnées de points comme informations visuelles dans la boucle de commande [Comport 03]. Cette approche consiste à pondérer chaque point par un poids w_i calculé à partir des résidus entre points de l'image courante et ceux de l'image désirée. Les résultats obtenus dans ce travail peuvent être directement exploités par notre approche utilisant les moments comme informations visuelles. En effet, il est possible de définir les moments calculés à partir

d'un ensemble de points en pondérant chaque point par la formule suivante :

$$m_{ij} = \sum_{h=0}^N w_i x_h^i y_h^j \quad (6.1)$$

Tous les résultats obtenus en utilisant la formule :

$$m_{ij} = \sum_{h=0}^N x_h^i y_h^j \quad (6.2)$$

restent valables si on utilise (6.1). L'utilisation des poids dans la définition des moments permettra de rejeter les erreurs dues aux points aberrants. Cependant, le calcul du poids à attribuer à chaque point se fait à partir des résidus entre points appariés. Il sera donc intéressant de trouver d'autres critères pour le calcul des poids w_i qui ne nécessitent pas un appariement entre les points.

Planification de trajectoire et problème d'occultation d'objet

Le découplage réalisé à l'aide des moments invariants a permis d'obtenir un bon comportement 3D du système tout en réalisant la commande dans l'image. Ces résultats peuvent être complétés en faisant appel à des techniques de planification comme celles proposées dans [Mezouar 02, Mezouar 03]. Le but sera d'assurer une convergence globale quelle que soit la position initiale de la caméra. Pour réaliser une telle tâche, il faut éviter principalement les deux problèmes suivants :

- les problèmes d'occultation d'un éventuel obstacle ou qu'une partie de l'objet quitte le champ de vision de la caméra pendant la réalisation de la tâche.
- les singularités de la matrice d'interaction ainsi que les minimums locaux

Dans le cas où on dispose d'un modèle de l'objet, il est possible de déterminer la pose de l'objet par rapport au repère caméra en utilisant les deux méthodes proposées dans le chapitre 4. Dans ce cas, la tâche de planification consisterait simplement à diviser ce déplacement 3D en petits déplacements tels que l'objet reste dans le champ de vision de la caméra et tels que la matrice d'interaction associée aux informations 2D ne soit pas singulière. Les informations visuelles 2D intermédiaires peuvent être déterminées à partir du modèle et des poses intermédiaires. Néanmoins, la connaissance du modèle de l'objet n'est pas nécessaire pour la réalisation de la tâche de planification. En effet, il est possible d'obtenir une reconstruction partielle de la pose (c'est-à-dire la matrice de rotation et le vecteur de translation à un facteur d'échelle près) en connaissant seulement les moments de l'image de l'objet dans une position telle que le plan image et le plan objet soient parallèles. Dans ce cas également, les méthodes proposées dans [Mezouar 02, Mezouar 03] peuvent être appliquées pour améliorer le comportement du système.

Pour éviter que l'objet quitte le champ de vision de la caméra, on peut également envisager d'utiliser des capteurs de vision omnidirectionnelle. En effet, ces derniers

permettent d'avoir un champ de vision plus large. Par contre, il est nécessaire d'adapter les résultats obtenus dans le cas de la projection perspective au modèle d'une caméra omnidirectionnelle.

L'ensemble des résultats développés dans ce travail concernent des objets plans. Il est évidemment souhaitable de généraliser ces résultats aux cas des objets non plans.

Annexe A

Invariants aux mouvements de rotation

Dans la suite, on donne une liste exhaustive des polynômes invariants $q(\mathbf{m}_p^n)$ aux rotations dans le cas d'un espace de dimension 2 et dans celui d'un espace de dimension 3 jusqu'aux moments d'ordre 5. Les polynômes sont tout d'abord donnés sous forme d'un vecteur $\boldsymbol{\alpha}$ tel que :

$$q(\mathbf{m}_p^n) = \boldsymbol{\alpha}^T \mathbf{m}_p^n$$

puis sous forme polynômiale dans le cas où la formule du polynôme correspondant est simple.

On note \mathbf{m}_p^1 le vecteur qui contient tous les moments d'ordre p définis dans un espace de dimension k . Il est construit de la manière suivante :

```
s = 1
pour p1 = p jusqu'à 0
pour p2 = p - p1 jusqu'à 0
:
pour pk-1 = p - p1 - ... - pk-2 jusqu'à 0
pk = p - p1 - ... - pk-1
mp1(s) = mp1...pk
s = s + 1
fin pour
:
fin pour
fin pour
```

où $\mathbf{m}_p^1(s)$ est la $s^{\text{ième}}$ composante du vecteur \mathbf{m}_p^1 .

Exemple :

Le vecteur \mathbf{m}_2^1 qui contient tous les moments d'ordre 2 définis dans un espace 2D est construit de la façon suivante :

$$\begin{aligned}
 s &= 1 \\
 p_1 &= 2 \\
 p_2 &= 2 - p_1 = 0 \\
 \mathbf{m}_p^1(1) &= m_{p_1 p_2} = m_{20} \\
 s &= 1 + 1 = 2 \\
 p_1 &= 1 \\
 p_2 &= 2 - p_1 = 1 \\
 \mathbf{m}_p^1(2) &= m_{p_1 p_2} = m_{11} \\
 s &= 2 + 1 = 3 \\
 p_1 &= 0 \\
 p_2 &= 2 - p_1 = 2 \\
 \mathbf{m}_p^1(3) &= m_{p_1 p_2} = m_{02}
 \end{aligned}$$

Le vecteur \mathbf{m}_2^1 est donc donné par :

$$\mathbf{m}_2^1 = (m_{20}, m_{11}, m_{02})$$

On note maintenant \mathbf{m}_p^n le vecteur qui contient les monômes d'ordre n calculés à partir des moments d'ordre p . Ce vecteur est construit de la façon suivante :

$$\begin{aligned}
 s &= 1 \\
 \max &= \dim(\mathbf{m}_p^1) \\
 \text{pour } j_1 &= n \text{ jusqu'à } 0 \\
 \text{pour } j_2 &= n - j_1 \text{ jusqu'à } 0 \\
 &\vdots \\
 \text{pour } j_{\max-1} &= n - j_1 - \dots - j_{\max-2} \text{ jusqu'à } 0 \\
 j_{\max} &= n - j_1 - \dots - j_{\max-1} \\
 \mathbf{m}_p^n(s) &= \prod_{i=0}^{i=\max} \mathbf{m}_p^1(j)^{j_i} \\
 s &= s + 1 \\
 \text{fin pour} & \\
 &\vdots \\
 \text{fin pour} & \\
 \text{fin pour} &
 \end{aligned}$$

Exemple :

Le vecteur \mathbf{m}_2^2 qui contient tous les moments d'ordre 2 définis dans un espace 2D est construit de la façon suivante :

$$\begin{aligned}
s &= 1 \\
max &= dim(\mathbf{m}_2^1) = 3 \\
j_1 &= 2 \\
j_2 &= 2 - j_1 = 0 \\
j_3 &= 2 - j_1 - j_2 = 0 \\
\mathbf{m}_p^n(1) &= \mathbf{m}_2^1(1)^{j_1} \mathbf{m}_2^1(2)^{j_2} \mathbf{m}_2^1(3)^{j_3} = m_{20}^2 \\
s &= s + 1 = 2 \\
j_1 &= 1 \\
j_2 &= 2 - j_1 = 1 \\
j_3 &= 2 - j_1 - j_2 = 0 \\
\mathbf{m}_p^n(2) &= \mathbf{m}_2^1(1)^{j_1} \mathbf{m}_2^1(2)^{j_2} \mathbf{m}_2^1(3)^{j_3} = m_{20}m_{11} \\
s &= s + 1 = 3 \\
j_1 &= 1 \\
j_2 &= 2 - j_1 - 1 = 0 \\
j_3 &= 2 - j_1 - j_2 = 1 \\
\mathbf{m}_p^n(3) &= \mathbf{m}_2^1(1)^{j_1} \mathbf{m}_2^1(2)^{j_2} \mathbf{m}_2^1(3)^{j_3} = m_{20}m_{02} \\
s &= s + 1 = 4 \\
j_1 &= 0 \\
j_2 &= 2 - j_1 = 2 \\
j_3 &= 2 - j_1 - j_2 = 0 \\
\mathbf{m}_p^n(4) &= \mathbf{m}_2^1(1)^{j_1} \mathbf{m}_2^1(2)^{j_2} \mathbf{m}_2^1(3)^{j_3} = m_{11}^2 \\
s &= s + 1 = 5 \\
j_1 &= 0 \\
j_2 &= 2 - j_1 = 1 \\
j_3 &= 2 - j_1 - j_2 = 1 \\
\mathbf{m}_p^n(5) &= \mathbf{m}_2^1(1)^{j_1} \mathbf{m}_2^1(2)^{j_2} \mathbf{m}_2^1(3)^{j_3} = m_{11}m_{02} \\
s &= s + 1 = 6 \\
j_1 &= 0 \\
j_2 &= 2 - j_1 - 1 = 0 \\
j_3 &= 2 - j_1 - j_2 = 2 \\
\mathbf{m}_p^n(6) &= \mathbf{m}_2^1(1)^{j_1} \mathbf{m}_2^1(2)^{j_2} \mathbf{m}_2^1(3)^{j_3} = m_{02}^2
\end{aligned}$$

Le vecteur \mathbf{m}_2^2 est donc donné par :

$$\mathbf{m}_2^2 = (m_{20}^2, m_{20}m_{11}, m_{20}m_{02}, m_{11}^2, m_{11}m_{02}, m_{02}^2)$$

Dans la section suivante, nous donnons une liste des moments invariants aux mouvements de rotation obtenus dans l'espace 2D.

A.1 Moments 2D

– polynôme d'ordre 1, moments d'ordre 2 :

$$q(\mathbf{m}_2^1) = m_{20} + m_{02} \quad (\text{A.1})$$

– polynôme d'ordre 2, moments d'ordre 2 :

$$q_1(\mathbf{m}_2^2) = m_{20}m_{02} - m_{11}^2 \quad (\text{A.2})$$

$$q_2(\mathbf{m}_2^2) = m_{20}^2 + 2m_{20}m_{02} + m_{02}^2 \quad (\text{A.3})$$

– polynômes d'ordre 1, moments d'ordre 3 :

L'ensemble est vide.

– polynômes d'ordre 2, moments d'ordre 3 :

$$\begin{aligned} \boldsymbol{\alpha}_1^T &= [0 \ 0 \ -1 \ 0 \ 1 \ 0 \ -1 \ 1 \ 0 \ 0] \\ q_1(\mathbf{m}_3^2) &= -m_{30}m_{12} + m_{21}^2 - m_{21}m_{03} + m_{12}^2 \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} \boldsymbol{\alpha}_2^T &= [1 \ 0 \ 3 \ 0 \ 0 \ 0 \ 3 \ 0 \ 0 \ 1] \\ q_2(\mathbf{m}_3^2) &= m_{30}^2 + 3m_{30}m_{12} + 3m_{21}m_{03} + m_{03}^2 \end{aligned} \quad (\text{A.5})$$

– polynômes d'ordre 3, moments d'ordre 3 :

L'ensemble est vide.

– polynômes d'ordre 4, moments d'ordre 3 :

$$\begin{aligned} \boldsymbol{\alpha}_1^T &= [0_{1 \times 9} \ -1 \ 0_{1 \times 4} \ 6 \ 0 \ -4 \ 0_{1 \times 5} \ -4 \ 3 \ 0_{1 \times 11}] \\ q_1(\mathbf{m}_3^4) &= -m_{30}^2m_{03}^2 + 6m_{30}m_{21}m_{12}m_{03} - 4m_{30}m_{12}^3 - 4m_{21}^3m_{03} + 3m_{21}^2m_{12}^2 \end{aligned} \quad (\text{A.6})$$

$$\begin{aligned} \boldsymbol{\alpha}_2^T &= [0_{1 \times 7} \ 3 \ 0 \ 2 \ 0 \ -6 \ 0_{1 \times 3} \ -6 \ 0 \ 2 \ 0_{1 \times 3} \ 3 \ 0 \ 2 \ 0_{1 \times 2} \\ &\quad 3 \ 0 \ -6 \ 0_{1 \times 2} \ 3 \ 0_{1 \times 4}] \\ q_2(\mathbf{m}_3^4) &= 3m_{30}^2m_{12}^2 + 2m_{30}^2m_{03}^2 - 6m_{30}m_{21}^2m_{12} - 6m_{30}m_{21}m_{12}m_{03} + 2m_{30}m_{12}^3 \\ &\quad + 3m_{21}^4 + 2m_{21}^3m_{03} + 3m_{21}^2m_{03}^2 - 6m_{21}m_{12}^2m_{03} + 3m_{12}^4 \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} \boldsymbol{\alpha}_3^T &= [0_{1 \times 3} \ -1 \ 0 \ 3 \ 0_{1 \times 4} \ -2 \ 0 \ -3 \ 6 \ 0_{1 \times 3} \ 3 \ 0 \ 1 \\ &\quad 0 \ -3 \ 0_{1 \times 2} \ -6 \ 0 \ 3 \ 0 \ -3 \ 0_{1 \times 2} \ 2 \ 0_{1 \times 3}] \\ q_3(\mathbf{m}_3^4) &= -m_{30}^3m_{03} + 3m_{30}^2m_{21}m_{12} - 2m_{30}m_{21}^3 - 3m_{30}m_{21}^2m_{03} + 6m_{30}m_{21}m_{12}^2 \\ &\quad + 3m_{30}m_{12}^2m_{03} + m_{30}m_{03}^3 - 3m_{21}^3m_{12} - 6m_{21}^2m_{12}m_{03} + 3m_{21}m_{12}^3 - 3m_{21}m_{12}m_{03}^2 \\ &\quad + 2m_{12}^3m_{03} \end{aligned} \quad (\text{A.8})$$

$$\begin{aligned}
\boldsymbol{\alpha}_4^T &= [0_{1 \times 2} \quad -1 \quad 0 \quad 1 \quad 0 \quad -1 \quad -2 \quad 0_{1 \times 3} \quad 3 \quad 0_{1 \times 2} \quad -6 \quad 0 \quad 3 \quad 0 \\
&\quad -1 \quad 0_{1 \times 3} \quad 3 \quad 0_{1 \times 2} \quad -2 \quad 0 \quad 3 \quad 0 \quad -1 \quad 0_{1 \times 2} \quad 1 \quad 0_{1 \times 2} \quad] \\
q_4(\mathbf{m}_3^4) &= -m_{30}^3 m_{12} + m_{30}^2 m_{21}^2 - m_{30}^2 m_{21} m_{03} - 2m_{30}^2 m_{12}^2 + 3m_{30} m_{21}^2 m_{12} \\
&\quad - 6m_{30} m_{21} m_{12} m_{03} + 3m_{30} m_{12}^3 - m_{30} m_{12} m_{03}^2 + 3m_{21}^3 m_{03} - 2m_{21}^2 m_{03}^2 \\
&\quad + 3m_{21} m_{12}^2 m_{03} - m_{21} m_{03}^3 + m_{12}^2 m_{03}^2
\end{aligned} \tag{A.9}$$

$$\begin{aligned}
\boldsymbol{\alpha}_5^T &= [\quad 1 \quad 0 \quad 6 \quad 0_{1 \times 3} \quad 6 \quad 9 \quad 0 \quad 2 \quad 0_{1 \times 4} \quad 18 \quad 0_{1 \times 3} \quad 6 \quad 0_{1 \times 6} \quad 9 \quad 0_{1 \times 3} \quad 6 \\
&\quad 0_{1 \times 4} \quad 1 \quad] \\
q_5(\mathbf{m}_3^4) &= m_{30}^4 + 6m_{30}^3 m_{12} + 6m_{30}^2 m_{21} m_{03} + 9m_{30}^2 m_{12}^2 + 2m_{30}^2 m_{03}^2 \\
&\quad + 18m_{30} m_{21} m_{12} m_{03} + 6m_{30} m_{12} m_{03}^2 + 9m_{21}^2 m_{03}^2 + 6m_{21} m_{03}^3 + m_{03}^4
\end{aligned} \tag{A.10}$$

À partir des polynômes d'ordre 2, on peut obtenir trois polynômes d'ordre 4, donc trois des cinq polynômes d'ordres 4 sont inutiles. On peut facilement montrer que les polynômes $\boldsymbol{\alpha}_1^T$, $\boldsymbol{\alpha}_4^T$ et $\boldsymbol{\alpha}_5^T$ sont à éliminer.

– polynômes d'ordre 1, moments d'ordre 4 :

$$\begin{aligned}
\boldsymbol{\alpha}_1^T &= [\quad 1 \quad 0 \quad 2 \quad 0 \quad 1 \quad] \\
q(\mathbf{m}_4^1) &= m_{40} + 2m_{22} + m_{04}
\end{aligned} \tag{A.11}$$

– polynômes d'ordre 2, moments d'ordre 4 :

$$\begin{aligned}
\boldsymbol{\alpha}_1^T &= [\quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad -4 \quad 0 \quad 3 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad] \\
q_1(\mathbf{m}_4^2) &= m_{40} m_{04} - 4m_{31} m_{13} + 3m_{22}^2
\end{aligned} \tag{A.12}$$

$$\begin{aligned}
\boldsymbol{\alpha}_2^T &= [\quad 0 \quad 0 \quad -3 \quad 0 \quad -2 \quad 3 \quad 0 \quad 2 \quad 0 \quad 0 \quad 0 \quad -3 \quad 3 \quad 0 \quad 0 \quad] \\
q_2(\mathbf{m}_4^2) &= -3m_{40} m_{22} - 2m_{40} m_{04} + 3m_{31}^2 + 2m_{31} m_{13} - 3m_{22} m_{04} + 3m_{13}^2
\end{aligned} \tag{A.13}$$

$$\begin{aligned}
\boldsymbol{\alpha}_3^T &= [\quad 3 \quad 0 \quad 12 \quad 0 \quad 2 \quad 0 \quad 0 \quad 16 \quad 0 \quad 0 \quad 0 \quad 12 \quad 0 \quad 0 \quad 3 \quad] \\
q_3(\mathbf{m}_4^2) &= 3m_{40}^2 + 12m_{40} m_{22} + 2m_{40} m_{04} + 16m_{31} m_{31} + 12m_{22} m_{04} + 3m_{04}^2
\end{aligned} \tag{A.14}$$

Un des trois polynômes précédent est à éliminer.

Dans la section suivante, nous donnons une liste de moments invariants aux mouvements de rotation obtenus dans l'espace 3D.

A.2 Moments 3D

– polynômes d'ordre 1, moments d'ordre 2 :

$$\begin{aligned} \boldsymbol{\alpha}_1^T &= [1 \ 0 \ 0 \ 1 \ 0 \ 1] \\ q(\mathbf{m}_2^1) &= m_{200} + m_{020} + m_{002} \end{aligned} \quad (\text{A.15})$$

– polynômes d'ordre 2, moments d'ordre 2 :

$$\begin{aligned} \boldsymbol{\alpha}_1^T &= [0_{1 \times 3} \ -1 \ 0 \ -1 \ 1 \ 0_{1 \times 4} \ 1 \ 0_{1 \times 5} \ -1 \ 1 \ 0_{1 \times 2}] \\ q_1(\mathbf{m}_2^2) &= -m_{200}m_{020} - m_{200}m_{002} + m_{110}^2 + m_{101}^2 - m_{020}m_{002} + m_{011}^2 \end{aligned} \quad (\text{A.16})$$

$$\begin{aligned} \boldsymbol{\alpha}_2^T &= [1 \ 0_{1 \times 2} \ 2 \ 0 \ 2 \ 0_{1 \times 9} \ 1 \ 0 \ 2 \ 0_{1 \times 2} \ 1] \\ q_2(\mathbf{m}_2^2) &= m_{200}^2 + 2m_{200}m_{020} + 2m_{200}m_{002} + m_{020}^2 + 2m_{020}m_{002} + m_{002}^2 \end{aligned} \quad (\text{A.17})$$

– polynômes d'ordre 3, moments d'ordre 2 :

$$\boldsymbol{\alpha}_1^T = [0_{1 \times 17} \ -1 \ 1 \ 0_{1 \times 6} \ 1 \ 0_{1 \times 2} \ -2 \ 0_{1 \times 8} \ 1 \ 0_{1 \times 18}] \quad (\text{A.18})$$

$$\begin{aligned} \boldsymbol{\alpha}_2^T &= [0_{1 \times 3} \ -1 \ 0 \ -1 \ 1 \ 0_{1 \times 4} \ 1 \ 0_{1 \times 3} \ -1 \ 0 \ -2 \ 0_{1 \times 2} \ -1 \\ &\quad 0_{1 \times 2} \ 1 \ 0_{1 \times 4} \ 2 \ 0_{1 \times 10} \ 1 \ 0_{1 \times 8} \ -1 \ 1 \ 0 \ -1 \ 0 \ 1 \\ &\quad 0_{1 \times 2} \] \end{aligned} \quad (\text{A.19})$$

$$\begin{aligned} \boldsymbol{\alpha}_3^T &= [1 \ 0_{1 \times 2} \ 3 \ 0 \ 3 \ 0_{1 \times 9} \ 3 \ 0 \ 6 \ 0_{1 \times 2} \ 3 \ 0_{1 \times 25} \ 1 \ 0 \ 3 \\ &\quad 0_{1 \times 2} \ 3 \ 0_{1 \times 3} \ 1 \] \end{aligned} \quad (\text{A.20})$$

Deux de ces polynômes sont à éliminer.

– polynômes d'ordre 1, moments d'ordre 3 :

l'ensemble est vide.

– polynômes d'ordre 2, moments d'ordre 3 :

$$\begin{aligned} \boldsymbol{\alpha}_1^T &= [0_{1 \times 3} \ -1 \ 0 \ -1 \ 0_{1 \times 4} \ 1 \ 0_{1 \times 4} \ -1 \ 0 \ -1 \ 0 \ 1 \ 0_{1 \times 4} \\ &\quad -1 \ 0 \ -1 \ 1 \ 0 \ -1 \ 0_{1 \times 4} \ 3 \ 0_{1 \times 5} \ 1 \ 0_{1 \times 6} \ -1 \ 0 \\ &\quad 1 \ 0 \ -1 \ 1 \ 0_{1 \times 2} \] \\ q(\mathbf{m}_3^2) &= -m_{300}m_{120} - m_{300}m_{102} + m_{210}^2 - m_{210}m_{030} - m_{210}m_{012} \\ &\quad + m_{201}^2 - m_{201}m_{021} - m_{201}m_{003} + m_{120}^2 - m_{120}m_{102} + 3m_{111}^2 \\ &\quad + m_{102}^2 - m_{030}m_{012} + m_{021}^2 - m_{021}m_{003} + m_{012}^2 \end{aligned} \quad (\text{A.21})$$

$$\begin{aligned}
\boldsymbol{\alpha}_2^T &= [\quad 1 \quad 0_{1 \times 2} \quad 3 \quad 0 \quad 3 \quad 0_{1 \times 9} \quad 3 \quad 0 \quad 3 \quad 0_{1 \times 6} \quad 3 \quad 0 \quad 3 \quad 0_{1 \times 2} \quad 3 \\
&\quad 0_{1 \times 4} \quad -3 \quad 0 \quad 0_{1 \times 9} \quad 1 \quad 0 \quad 3 \quad 0_{1 \times 3} \quad 3 \quad 0_{1 \times 2} \quad 1 \quad] \\
q(\mathbf{m}_3^2) &= m_{300}^2 + 3m_{300}m_{120} + 3m_{300}m_{102} + 3m_{210}m_{030} + 3m_{210}m_{012} \\
&+ 3m_{201}m_{021} + 3m_{201}m_{003} + 3m_{120}m_{102} - 3m_{111}^2 + m_{030}^2 + 3m_{030}m_{012} \\
&+ 3m_{021}m_{003} + m_{003}^2
\end{aligned} \tag{A.22}$$

– polynômes d'ordre 1, moments d'ordre 4 :

$$\begin{aligned}
\boldsymbol{\alpha}_1^T &= [\quad 1 \quad 0 \quad 0 \quad 2 \quad 0 \quad 2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 2 \quad 0 \quad 1 \quad] \\
q(\mathbf{m}_4^1) &= m_{400} + 2m_{220} + 2m_{202} + m_{040} + 2m_{022} + m_{004}
\end{aligned} \tag{A.23}$$

– polynômes d'ordre 2, moments d'ordre 4 :

$$\begin{aligned}
\boldsymbol{\alpha}_1^T &= [\quad 0_{1 \times 10} \quad 1 \quad 0 \quad 2 \quad 0 \quad 1 \quad 0_{1 \times 5} \quad -4 \quad 0 \quad -4 \quad 0_{1 \times 11} \\
&\quad -4 \quad 0 \quad -4 \quad 0_{1 \times 5} \quad 3 \quad 0 \quad 2 \quad 0_{1 \times 6} \quad 2 \quad 0 \quad 2 \\
&\quad 4 \quad 0_{1 \times 6} \quad -4 \quad 0 \quad -4 \quad 0 \quad 3 \quad 0_{1 \times 4} \quad 2 \quad 0 \quad 2 \\
&\quad 0_{1 \times 4} \quad -4 \quad 0_{1 \times 6} \quad 4 \quad 0 \quad -4 \quad 0_{1 \times 5} \quad 4 \quad 0_{1 \times 16} \quad 1 \quad 0_{1 \times 2} \\
&\quad -4 \quad 0 \quad 3 \quad 0_{1 \times 5} \quad]
\end{aligned} \tag{A.24}$$

$$\begin{aligned}
\boldsymbol{\alpha}_2^T &= [\quad 0_{1 \times 3} \quad -3 \quad 0 \quad -3 \quad 0_{1 \times 4} \quad -2 \quad 0 \quad -4 \quad 0 \quad -2 \quad 3 \quad 0_{1 \times 4} \\
&\quad 2 \quad 0 \quad 2 \quad 0_{1 \times 6} \quad 3 \quad 0_{1 \times 4} \quad 2 \quad 0 \quad 2 \quad 0_{1 \times 7} \quad -7 \quad 0_{1 \times 4} \\
&\quad -3 \quad 0 \quad -7 \quad 0 \quad -4 \quad 7 \quad 0_{1 \times 6} \quad 2 \quad 0 \quad 2 \quad 0_{1 \times 6} \quad -4 \\
&\quad 0 \quad -7 \quad 0 \quad -3 \quad 3 \quad 0 \quad 2 \quad 0_{1 \times 6} \quad 7 \quad 0 \quad 2 \quad 0_{1 \times 5} \\
&\quad 7 \quad 0_{1 \times 6} \quad 3 \quad 0_{1 \times 7} \quad -3 \quad 0 \quad -2 \quad 3 \quad 0 \quad 2 \quad 0_{1 \times 3} \quad -3 \\
&\quad 3 \quad 0_{1 \times 2} \quad]
\end{aligned} \tag{A.25}$$

$$\begin{aligned}
\boldsymbol{\alpha}_3^T &= [\quad 3 \quad 0_{1 \times 2} \quad 12 \quad 0 \quad 12 \quad 0_{1 \times 4} \quad 2 \quad 0 \quad 4 \quad 0 \quad 2 \quad 0_{1 \times 5} \\
&\quad 16 \quad 0 \quad 16 \quad 0_{1 \times 11} \quad 16 \quad 0 \quad 16 \quad 0_{1 \times 7} \quad 16 \quad 0_{1 \times 4} \quad 12 \quad 0 \\
&\quad 16 \quad 0 \quad 4 \quad -16 \quad 0_{1 \times 6} \quad 16 \quad 0 \quad 16 \quad 0_{1 \times 6} \quad 4 \quad 0 \quad 16 \\
&\quad 0 \quad 12 \quad 0_{1 \times 2} \quad 16 \quad 0_{1 \times 6} \quad -16 \quad 0 \quad 16 \quad 0_{1 \times 5} \quad -16 \quad 0_{1 \times 12} \quad 3 \\
&\quad 0 \quad 12 \quad 0 \quad 2 \quad 0_{1 \times 2} \quad 16 \quad 0_{1 \times 3} \quad 12 \quad 0_{1 \times 2} \quad 3 \quad]
\end{aligned} \tag{A.26}$$

– polynômes d'ordre 1, moments d'ordre 5 :
L'ensemble est vide

Annexe B

Minima, zone d'influence, ligne de partage des eaux

B.1 Définitions

Dans cette annexe, nous rappelons les notions de minima, de zone d'influence et de squelette par zone d'influence géodésique.

B.1.1 Minima

Un minimum m d'une image I à l'altitude h est un plateau connexe de pixels de valeur h d'où il est impossible d'atteindre un point d'altitude inférieure sans avoir à monter. Dans notre cas, il est nécessaire que le minimum soit formé d'un seul point et non d'un plateau.

B.1.2 Distance géodésique

Soit A un ensemble connexe. On définit la distance géodésique dans A par

$$d_A(x, y) = \inf \ell(P), P \text{ chemin reliant } x \text{ à } y \text{ inclus dans } A$$

$\ell(P)$ est égal au longueur du chemin P .

B.1.3 Zone d'influence géodésique

Supposons que A contienne k minima B_1, \dots, B_k . Soit $B = \bigcup_{i=1}^k B_i$. La zone d'influence géodésique d'un minimum B_i dans l'ensemble A , noté $iz_A(B_i)$, est le lieu des points de A dont la distance géodésique à B_i est inférieure à leur distance à tout autre minimum (voir la figure B.1):

$$iz_A(B_i) = \{p \in A \mid \forall j \in 1..k, j \neq i, d_A(p, B_i) < d_A(p, B_j)\}$$

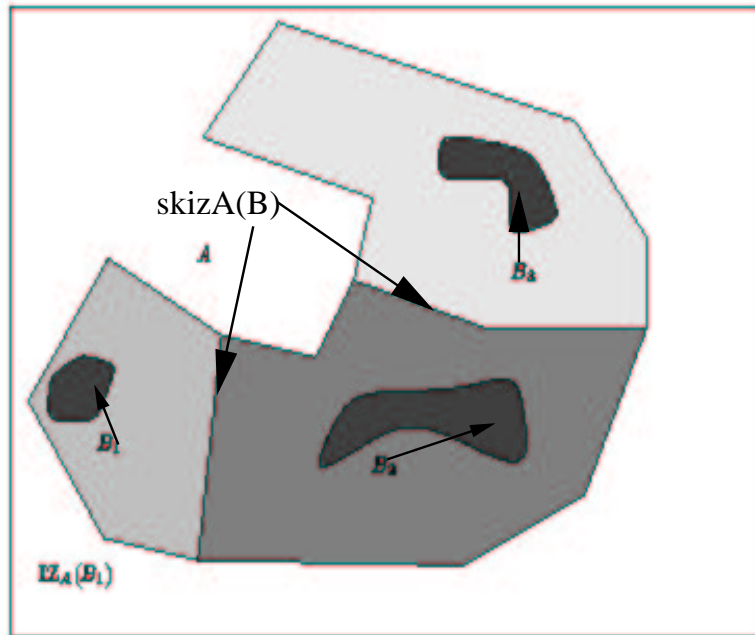


FIG. B.1 – Zone d'influence géodésique

B.1.4 Squelette par zone d'influence géodésique

Les pixels de A qui n'appartiennent à aucune zone d'influence constituent le squelette par zone d'influence géodésique de B dans A , noté $skiz_A(B)$:

$$skiz_A(B) = A - IZ_A(B) \text{ avec } IZ_A(B) = \bigcup_{i=1}^k iz_A(B_i)$$

ils déterminent par conséquent la frontière de la zone d'influence associée à chaque minimum.

Bibliographie

- [Abu-Mustapha 85] Y. S. Abu-Mustapha & D. Psaltis. *Image Normalisation by Complex Moments*. IEEE Trans. on PAMI, vol. 7, no. 1, pages 46–55, 1985.
- [Belkassim 90] S. O. Belkassim, M. Shridhar & M. Ahmadi. *Shape-Contour Recognition Using Moment Invariants*. In 10th Int. Conf. on Pattern Recognition, pages 649–651, Atlantic City, NJ, USA, Juin 1990.
- [Bellman 61] R. Bellman. *Adaptive control processes: A guided tour*. Princeton University Press, 1961.
- [Berrani 02] S. A. Berrani, L. Amsaleg & P. Gros. *Recherche par similarité dans les bases de données multidimensionnelles: panorama des techniques d'indexation*. RSTI - Ingénierie des systèmes d'information. Bases de données et multimédia, vol. 7, no. 5-6, pages 9–44, 2002.
- [Beyer 92] H. Beyer. *Accurate calibration of CCD cameras*. In IEEE Int. Conf. on Computer Vision and Pattern Recognition, CV-PR'94, pages 96–101, Champaign, Illinois, Juin 1992.
- [Boukir 93] S. Boukir. *Reconstruction 3D d'un environnement statique à l'aide d'une caméra mobile commandable*. PhD thesis, Université de Rennes 1, IRISA, Octobre 1993.
- [Chaumette 90] F. Chaumette. *La relation vision-commande: théorie et applications à des tâches robotiques*. PhD thesis, Université de Rennes 1, IRISA, Juillet 1990.
- [Chaumette 96] F. Chaumette, S. Boukir, P. Bouthemy & D. Juvin. *Structure from Controlled Motion*. IEEE Trans. on PAMI, vol. 18, pages 492–504, 1996.
- [Chaumette 98] F. Chaumette. *Potential problems of stability and convergence in image-based and position-based visual servoing*. In D. Kriegman, G. Hager & A.S. Morse, editeurs, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag, 1998.
- [Chaumette 00] F. Chaumette & E. Malis. *2 1/2D visual servoing: A possible solution to improve image-based and position-based vi-*

- sual servoings*. In IEEE Int. Conf. on Robotics and Automation, ICRA'00, volume 1, pages 630–635, Mai 2000.
- [Chaumette 02a] F. Chaumette. *Asservissement visuel*. In W. Khalil, éditeur, La commande des robots manipulateurs, Traité IC2, chapitre 3, pages 105–150. Hermès, 2002.
- [Chaumette 02b] F. Chaumette. *A first step toward visual servoing using image moments*. In IEEE/RSJ IROS'02, pages 378–383, Lausanne, Suisse, Octobre 2002.
- [Chaumette 04] F. Chaumette. *Image moments: a general and useful set of features for visual servoing*. IEEE Trans. on Robotics and Automation, à paraître 2004.
- [Chesi 02] G. Chesi, K. Hashimoto, D. Prattichizzo & A. Vicino. *Keeping features in the camera's field of view: a visual servoing strategy*. In 15th Int. Symp. on Mathematical Theory of Networks and Systems, Notre-Dame, Indiana, 2002.
- [Chesi 03] G. Chesi, K. Hashimoto & D. Prattichizzo. *A switching control law for keeping features in the field of view in eye-in-hand visual servoing*. In IEEE Int. Conf. on Robotics and Automation, ICRA'03, pages 3911–3916, Taïpei, Septembre 2003.
- [Coiffet 80] P. Coiffet & P. Rives. *Reconnaissance par un robot de l'orientation d'objet tridimensionnels en vue de taches de saisie automatique*. R.A.I.R.O. Automatique/Systems Analysis and Control, vol. 14, no. 1, 1980.
- [Collewet 00] C. Collewet & F. Chaumette. *A contour approach for image-based control of objects with complex shape*. In IEEE/RSJ IROS'00, pages 751–756, Takamatsu, Japon, Novembre 2000.
- [Comport 03] A. Comport, M. Pressigout, E. Marchand & F. Chaumette. *A Visual Servoing Control Law that is Robust to Image Outliers*. In IEEE Int. Conf. on Intelligent Robots and Systems, IROS'03, volume 1, pages 492–497, Las Vegas, Nevada, Octobre 2003.
- [Corke 01] P. I. Corke & S. A. Hutchinson. *A New Partitioned Approach to Image-Based Visual Servo Control*. IEEE Trans. on Robotics and Automation, vol. 17, no. 4, pages 507–515, Août 2001.
- [Cowan 02] N. J. Cowan, J. D. Weingarten & D. E. Koditschek. *Visual Servoing Via Navigation Functions*. IEEE Trans. on Robotics and Automation, Août 2002.
- [Cyganski 85] D. Cyganski & J. A. Orr. *Application of Tensor Theory to Object Recognition and Orientation Determination*. IEEE Trans. on PAMI, vol. 7, pages 662–673, Novembre 1985.

- [Deguchi 00] K. Deguchi. *A Direct Interpretation of Dynamic Images with Camera and Object Motions for Vision Guided Robot Control*. Int. Journal of Computer Vision, vol. 37, no. 1, pages 7–20, 2000.
- [Dementhon 95] D. F. Dementhon & L. S. Davis. *Model-Based Object Pose in 25 Lines of Code*. Int. Journal of Computer Vision, vol. 15, no. 1-2, pages 123–141, Juin 1995.
- [Dhome 89] M. Dhome, M. Richetin, J. T. LaPresté & G. Rives. *Determination of the Attitude of 3-D Objects from a Single Perspective View*. IEEE Trans. on PAMI, vol. 11, no. 12, pages 1265–1278, Décembre 1989.
- [Dhome 90] M. Dhome, J. T. LaPresté, G. Rives & M. Richetin. *Spatial Localization of Modelled Objects of Revolution in Monocular Perspective Vision*. In European Conf. on Computer Vision, ECCV90, LNCS 427, pages 475–485, Antibes, Avril 1990.
- [Donoho 00] D. L. Donoho. *High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality*. In The American Mathematical Society Conference “Math Challenges of the 21st Century”, Août 2000.
- [Espiau 92] B. Espiau, F. Chaumette & P. Rives. *A new approach to visual servoing in robotics*. IEEE Trans. on Robotics and Automation, vol. 8, no. 3, pages 313–326, Juin 1992.
- [Feddema 89] J. Feddema & O. Mitchell. *Vision-guided servoing with feature-based trajectory generation*. IEEE Trans. on Robotics and Automation, vol. 5, pages 691–700, Octobre 1989.
- [Fermuller 00] C. Fermuller & Y. Aloimonos. *Observability of 3D motion*. Int. Journal of Computer Vision, vol. 37, no. 1, pages 43–64, Juin 2000.
- [Flusser 00] J. Flusser. *On the Independance of Rotation Moment Invariants*. Pattern Recognition, vol. 33, pages 1405–1410, 2000.
- [Gangloff 98] J. Gangloff, M. De Mathelin & G. Abba. *6 DOF hight speed dynamic visual servoing using GPC controllers*. In IEEE Int. Conf. on Robotics and Automation, ICRA’98, pages 2008–2013, Louvain, Belgique, Mai 1998.
- [Gangloff 99] J. Gangloff. *Asservissements visuels rapides d’un robot manipulateur à six degrés de liberté*. PhD thesis, Université Louis Pasteur de Strasbourg, Janvier 1999.
- [Gans 03] N. R. Gans & S. A. Hutchinson. *An Asymptotically Stable Switched System Visual Controller for Eye in Hand Robots*. In IEEE/RSJ IROS’03, pages 735–742, Las Vegas, Nevada USA, Octobre 2003.

- [Guerivich 64] G. B. Guerivich. Foundations of the theory of algebraic invariants. Noordhoff LTD, 1964.
- [Hamel 00] T. Hamel & R. Mahony. *Robust Visual servoing for underactuated dynamic systems*. In IEEE Int. Conf. on Decision and Control, Sydney, Australia, 2000.
- [Haralick 89] R. Haralick, H. Joo, C. N. Lee, X. Zhuang, V. Vaidya & M. B. Kim. *Pose estimation from corresponding point data*. IEEE Trans. on Systems, Man and Cybernetics, vol. 19, no. 6, pages 1426–1446, 1989.
- [Harris 88] C. Harris & M. Stephens. *A combined corner and edge detector*. In In Proceeding of the 4th Alvey Vision Conference, pages 147–151, 1988.
- [Hashimoto 93] K. Hashimoto & H. Kimura. Lq optimal and non linear approaches to visual servoing, volume 7 of *World scientific Series in Robotics and Automated Systems*. World scientific press, Singapour, 1993.
- [Hashimoto 96] K. Hashimoto, T. Bienne & H. Kimura. *Visual Servoing with hand-eye manipulator-optimal control approach*. IEEE Trans. on Robotics and Automation, vol. 12, no. 5, pages 766–774, Octobre 1996.
- [Hayt 89] W. H. Hayt. Engineering electromagnetics. Mc Graw-Hill, 5eme edition, 1989.
- [Hilbert 93] Hilbert, D. Reinhard & C. Trad. Theory of algebraic invariant. Cambridge University, 1993.
- [Horaud 89] R. Horaud, B. Conio, O. Leboulleux & B. Lacolle. *An Analytic Solution For The Perspective 4-Point Problem*. Computer Vision, Graphics and Image Processing, vol. 47, pages 33–44, 1989.
- [Horaud 95] R. Horaud & O. Monga. Vision par ordinateur, outils fondamentaux. Hermès, Paris, 2ème edition, 1995.
- [Hu 62] M-K Hu. *Visual Pattern Recognition by Moments Invariants*. IRE Trans. on Information Theory, vol. 8, pages 179–187, Février 1962.
- [Kanatani 99] K. Kanatani & N. Ohta. *Accuracy bounds and optimal computation of homography for image*. In Proceedings of the 7th International Conference on Computer Vision, pages 73–78, Kerkyra, Greece, 1999.
- [Khadraoui 96] D. Khadraoui. *La commande référencée vision pour le guidage automatique des véhicules*. PhD thesis, Université Blaise Pascal de Clermont-Ferrand, LASMEA, Décembre 1996.
- [Khalil 99] W. Khalil & E. Dombre. Modélisation, identification, et commande des robots. Hermès, Paris, 2ème edition, 1999.

- [Lo 89] C. H. Lo & H. S. Don. *3D Moment Forms: Their Construction and Application to Object Identification and Positioning*. IEEE Trans. on PAMI, vol. 11, no. 10, pages 1053–1064, Octobre 1989.
- [Lowe 87] D. G. Lowe. *Three-Dimensional Object Recognition from Single Two-Dimensional Images*. Artificial Intelligence, vol. 31, no. 3, 1987.
- [Ma 93] S. D. Ma. *Conics-Based Stereo, Motion Estimation and Pose Determination*. Int. Journal of Computer Vision, vol. 10, no. 1, pages 7–25, Février 1993.
- [Mahony 02] R. Mahony, P. Corke & F. Chaumette. *Choice of image features for depth-axis control in image-based visual servo control*. In IEEE/RSJ IROS'02, pages 390–395, Lausanne, Suisse, Octobre 2002.
- [Malis 98] E. Malis. *Contributions à la modélisation et à la commande en asservissement visuel*. PhD thesis, l'Université de Rennes 1, IRISA, 1998.
- [Malis 99] E. Malis, F. Chaumette & S. Boudet. *2 1/2 D Visual Servoing*. IEEE Trans. on Robotics and Automation, vol. 15, no. 2, pages 238–250, Avril 1999.
- [Malis 04] E. Malis. *Improving vision-based control using efficient second-order minimization techniques*. In IEEE Int. Conf. on Robotics and Automation, ICRA'04, New Orleans, April 2004.
- [Mamistvalov 70] A.G. Mamistvalov. *On the Fundamental Theorem of Moment Invariants*. Bull. Acade. Sciences Georgian SSR, vol. 59, no. 2, pages 297–300, Août 1970. en russe.
- [Mamistvalov 98] A. G. Mamistvalov. *n-Dimensional Moment Invariants and Conceptual Theory of Recognition n-Dimensional Solids*. IEEE Trans. on PAMI, vol. 20, no. 8, pages 819–831, Août 1998.
- [Marchand 02] E. Marchand & F. Chaumette. *Virtual Visual Servoing: a framework for real-time augmented reality*. In H.-P. Seidel (eds.) G. Drettakis, editeur, in EUROGRAPHICS 2002 Conference Proceeding, Saarebrücke, Allemagne, Septembre 2002.
- [Markandey 92] V. Markandey & R. J. P. deFigueiredo. *Robot Sensing Techniques based on High-dimensional moment invariants and tensors*. IEEE Trans. on Robotics and Automation, vol. 8, no. 2, pages 186–195, Avril 1992.
- [Martinet 96] P. Martinet, J. Gallice & D. Khadraoui. *Vision-based control law using 3D visual features*. In Proc. of the Automations

- World Congress, volume 3, pages 497–502, WAC'96, Montpellier, 1996.
- [Martinet 97] P. Martinet, N. Daucher, J. Gallice & M. Dhome. *Robot control using monocular pose estimation*. In Workshop on New Trends in Image-based Robot Servoing, IROS'97, pages 1–12, Septembre 1997.
- [Mezouar 02] Y. Mezouar & F. Chaumette. *Path Planning For Robust Image-based Control*. IEEE Trans. on Robotics and Automation, vol. 18, no. 4, pages 534–549, Août 2002.
- [Mezouar 03] Y. Mezouar & F. Chaumette. *Optimal Camera Trajectory with Image-based Control*. Int. Journal of Robotics Research, vol. 22, no. 10, pages 781–804, Octobre 2003.
- [Michel 93] H. Michel & P. Rives. *Singularities in the determination of the situation of a robot effector from the perspective view of three points*. Rapport technique 1850, INRIA, 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*. In Int. Symp. on Experimental robot, LNCIS, volume 250, pages 99–108. Springer-Verlag, 2000.
- [Mukundan 92] R. Mukundan. *Estimation of quaternion parameters from two dimensional image moment*. Graphical Model and Image Processing, vol. 54, no. 4, pages 345–350, Juillet 1992.
- [Mukundan 96] R. Mukundan & K. R. Ramakrishnan. *An Iterative solution for object pose parameters using image moments*. Pattern Recognition Letters, vol. 17, pages 1279–1284, 1996.
- [Mukundan 98] R. Mukundan & K. R. Ramakrishnan. *Moment functions in image analysis theory and application*. World Scientific Publishing Co.Pte.Ltd, 1998.
- [Oberkampf 96] D. Oberkampf, D. F. Dementhon & L. S. Davis. *Iterative Pose Estimation using Coplanar Feature Points*. CVGIP: Image Understanding, vol. 63, no. 3, pages 495–511, 1996.
- [Papanikolopoulos 93] N. Papanikolopoulos, P. Khosla & T. Kanade. *Visual Tracking of a moving camera mounted on robot: a combination of control and vision*. IEEE Trans. on Robotics and Automation, vol. 9, no. 1, pages 14–35, Février 1993.
- [Papanikolopoulos 95] N. Papanikolopoulos. *Selection of features and evaluation of visual measurements during robotic visual servoing tasks*. Journal of Intelligent and Robotics Systems, 1995.
- [Phong 95] T. Q. Phong, R. Horaud, A. Yassine & P. D. Tao. *Object Pose from 2-D to 3-D Point and Line Correspondences*. Int. Journal of Computer Vision, vol. 15, no. 3, pages 225–243, 1995.

- [Prokop 92] R. J. Prokop & A. P. Reeves. *A Survey of Moments Based techniques for Unoccluded Object Representation*. Graphical models and Image Processing, vol. 54, no. 5, pages 438–460, Septembre 1992.
- [Reddi 81] S. S. Reddi. *Radial and Angular Moment Invariants for Image Identification*. IEEE Trans. on PAMI, vol. 3, no. 2, pages 240–242, 1981.
- [Reiss 91] T. H. Reiss. *The Revised Fundamental Theorem of Moment Invariants*. IEEE Trans. on PAMI, vol. 13, no. 8, pages 830–834, Août 1991.
- [Reyes 98] F. Reyes & R. Kelly. *Experimental evaluation of fixed-camera direct visual controllers on direct drive robot*. In IEEE Int. Conf. on Robotics and Automation, volume 3, pages 2327–2332, Louvain, Belgique, Mai 1998.
- [Robert 3] L. Robert. *Perception stéréoscopique de courbes et de surfaces tridimensionnelles. Application à la robotique mobile*. PhD thesis, l'Ecole Polytechnique, INRIA Sophia Antipolis, Mars 1993.
- [Safae-Rad 92] R. Safae-Rad, I. Tchoukanov, K.C. Smith & B. Benhabib. *Three-Dimensional Location Estimation of Circular Features for Machine Vision*. IEEE Trans. on Robotics and Automation, vol. 8, no. 5, pages 624–640, Octobre 1992.
- [Samson 91] C. Samson, M. Le Borgne & B. Espiau. *Robot control: The task function approach*. Clarendon Press, Oxford, 1991.
- [Shi 94] J. Shi & C. Tomasi. *Good features to track*. In CVPR'94, pages 593–600, Seattle, Juin 1994.
- [Sluzek 95] A. Sluzek. *Identification and inspection of 2-D objects using new moment-based shape descriptors*. Pattern Recognition Letters, vol. 16, pages 687–697, 1995.
- [Smith 94] C. Smith & N. Papanikolopoulos. *Computation of shape through controled active exploration*. In IEEE Int. Conf. on Robotics and Automation, ICRA'94, volume 3, pages 2516–2521, San Diego, Californie, Mai 1994.
- [Stewart 91] J. Stewart. *Calculus*. Brooks/cole, 2ème edition, 1991.
- [Stokely 88] T. L. Faber E. M. Stokely. *Orientation of 3-D Structures in medical images*. IEEE Trans. on PAMI, vol. 10, no. 5, pages 626–633, Septembre 1988.
- [Sundareswaran 98] V. Sundareswaran & R. Behringer. *Visual servoing based augmented reality*. In IEEE Int. Workshop on Augmented Reality, San Francisco, Novembre 1998.
- [Tahri 03] O. Tahri & F. Chaumette. *Application of moment invariants*

- to visual servoing*. In IEEE Int. Conf. on Robotics and Automation, ICRA'03, pages 4276–4281, Taïpei, Septembre 2003.
- [Tahri 04] O. Tahri & F. Chaumette. *Image Moments: Generic Descriptors for Decoupled Image-based Visual Servo*. In IEEE Int. Conf. on Robotics and Automation, ICRA'04, volume 2, pages 1185–1190, La Nouvelle-Orléans, Louisiane, USA, Avril 2004.
- [Teague 80] M.R. Teague. *Image Analysis via the General Theory of Moments*. Journal of Opt. Soc. of America, vol. 70, pages 920–930, Août. 1980.
- [Tsakiris 98] D. P. Tsakiris, P. Rives & C. Samson. *Extending visual servoing techniques to nonholonomic mobile robots*. In D. Kriegman, G. Hager & A. S. Morse, editeurs, The Confluence of Vision and Control, numéro 237, pages 106–117. Springer-Verlag, 1998. LNCIS.
- [VanGool 95] L. VanGool, T. Moonsand, E. Pauwelsand & A. Oosterlinck. *Vision and Lie's approach to invariance*. Image and Vision Computing, vol. 13, no. 4, pages 259–1995, Mai 1995.
- [Verleysen 03] M. Verleysen. *Learning high-dimensional data*. In S. Ablameyko V. Piuri M. Gori & L. Goras, editeurs, Limitations and future trends in neural computation, pages 141–162. IOS Press, 2003.
- [Vincent 92] L. Vincent & P. Soille. *Watersheds in digital spaces: An efficient Algorithm Based on Immersion Simulations*. IEEE Trans. on PAMI, vol. 13, no. 6, pages 583–598, Juin 1992.
- [Walsh 95] A. Walsh & O. Kübler. *Complete Sets of Complex Zernike Moments Invariants and the role of the Pseudo-invariants*. IEEE Trans. on PAMI, vol. 17, no. 11, pages 1106–1110, Novembre 1995.
- [Weiss 87] L. E. Weiss & A. C. Sanderson. *Dynamic sensor-based control of robots with visual feedback*. IEEE Journal of Robotics and Automation, vol. 3, no. 5, pages 404–417, Octobre 1987.
- [Wilson 96] W. Wilson, C. Hulls & G. Bell. *Relative end-effector control using cartesian position-based visual servoing*. IEEE Trans. on Robotics and Automation, vol. 12, no. 5, pages 684–696, Octobre 1996.
- [Wong 78] R. Y. Wong & E. L. Hall. *Scene Matching with Invariants Moments*. In Computer Graphics Image Proceeding, volume 8, pages 16–24, Septembre 1978.
- [Wong 95] W. H. Wong, W. C. Siu & K. M. Lam. *Generation of Moment Invariants and their uses for Characters Recognition*. Pattern Recognition, vol. 16, pages 115–123, 1995.

- [Xie 89] M. Xie. *Contribution à la vision dynamique*. PhD thesis, Université de Rennes 1, IRISA, Juin 1989.

Abstract

This thesis is concerned with the control of robot system using visual information. The classical techniques are based on simple geometrical features: point coordinates, straight lines, etc. The objects that can be treated using those features are thus limited. In this thesis, we propose the use of image moments to deal with objects with complex shapes and to obtain an adequate behavior of the system. The second part of the thesis deals with the pose estimation problem. More precisely, we determine the object position and its orientation with respect to the camera frame using the corresponding image. In this work, a new method using image moments and virtual visual servoing is proposed to solve this problem. Furthermore, another method based on an exhaustive optimization algorithm is proposed, which can be useful to initialize the method using virtual visual servoing. Experimental results using planar objects given by a continuous or discrete distributions are given to validate the above approaches.

Keywords : Vision, Robotics, Visual Servoing, Pose Estimation, Moments Invariants.

Résumé de thèse

La thématique de cette thèse s'inscrit dans le domaine de l'asservissement visuel qui consiste à contrôler les mouvements d'un système robotique en utilisant des informations fournies par un capteur de vision. Les techniques classiques sont basées sur des primitives géométriques particulières : coordonnées de point, équations de droites, équations d'ellipses, etc. Par conséquent, la famille des scènes concernées est limitée à des scènes très simples. Dans cette thèse, on propose d'utiliser les moments d'un objet dans l'image, ces derniers pouvant en effet caractériser des objets de forme beaucoup plus complexe.

Dans un premier temps, nous proposons une nouvelle méthode basée sur un calcul variationnel pour déterminer des formules analytiques de moments invariants. La méthode présentée permet de déterminer des moments invariants aux rotations, aux translations et aux changements d'échelle. Nous exploitons ensuite ces résultats pour choisir six informations visuelles afin contrôler les six degrés de liberté d'une caméra. L'objectif est de réaliser un schéma de commande découplé et de minimiser les non linéarités du système. L'ensemble de ces résultats a été validé en utilisant des objets de forme plane et définis soit par une distribution continue soit par un ensemble de points.

La deuxième partie de la thèse porte sur le calcul de pose. En disposant d'un modèle 3D de l'objet, il s'agit de reconstruire la position et l'orientation de cet objet dans le repère de la caméra en utilisant l'image correspondante. Notre approche consiste à commander virtuellement les mouvements de la caméra de telle sorte qu'à sa position finale, la projection du modèle CAO de l'objet corresponde à l'image acquise. Cette méthode est équivalente à une méthode non linéaire sujette à minima locaux. Pour résoudre ce problème, nous utilisons une méthode d'initialisation efficace basée sur le calcul d'une table bidimensionnelle. Nous proposons enfin une méthode de calcul de pose partielle n'utilisant pas la connaissance du modèle CAO de l'objet. Elle revient à utiliser les moments pour estimer la collinéation entre deux images d'un objet plan.

Mots-Clefs : Vision, Robotique, Asservissement Visuel, Calcul de Pose, Moments Invariants.