

No d'Ordre : 1074

THESE

Présentée devant

L'Université de Rennes I

U.F.R. Structure et Propriétés de la Matière

Pour obtenir

Le Titre de Docteur de l'Université de Rennes I
Mention Traitement du signal et Télécommunications

par

Samia BOUKIR

Titre de la thèse

Reconstruction 3D d'un environnement statique
par vision active

Soutenue le 21 Octobre 1993 devant la commission d'examen

MM.	Jean-Jacques	Fuchs	Président
	Michel	Dhome	Rapporteur
	Patrick	Rives	Rapporteur
	Patrick	Bouthemy	Examineur
	Serge	Castan	Examineur
	François	Chaumette	Examineur
	Didier	Juvin	Examineur
	Roger	Mohr	Examineur

Remerciements

Je tiens à remercier Monsieur Jean-Jacques FUCHS, Professeur à l'Université de Rennes 1, d'avoir accepté la présidence de mon jury de thèse.

Je remercie également :

Messieurs Michel DHOME, Chargé de Recherche au C.N.R.S Clermont Ferrand et Patrick RIVES, Chargé de Recherche à l'I.N.R.I.A. Sophia Antipolis, qui ont bien voulu assurer la charge de rapporteurs de ces travaux,

Monsieur Serge CASTAN, Professeur à l'Université Paul Sabatier de Toulouse,

Monsieur Roger MOHR, Professeur à l'I.N.P.G. de Grenoble

pour l'intérêt qu'ils portent à ce travail en acceptant de le juger.

Je remercie aussi très vivement :

François CHAUMETTE, Chargé de Recherche à l'I.N.R.I.A. Rennes,

Patrick BOUTHEMY, Directeur de Recherche à l'I.N.R.I.A. Rennes et

Didier JUVIN, Docteur-Ingénieur au CEA Saclay

pour leur nombreux conseils et suggestions constructives.

Mes remerciements vont également à tous ceux qui, à des titres divers, ont contribué à l'aboutissement de ce mémoire, et en particulier à Jean-Pierre Guérin et à Marc Viala du CEA Saclay.

Un petit mot particulier pour Farabi et pour Lila et toute mon amitié. Ma sympathie aussi à tous les membres de l'équipe TEMIS pour l'ambiance agréable qu'ils ont créée autour de moi, sans oublier les membres de l'équipe GAAP du DEIN-SLA CEA Saclay.

Table des Matières

Introduction	7
1 La vision tridimensionnelle	11
1.1 Introduction	11
1.2 Acquisition de données tridimensionnelles	12
1.2.1 Les capteurs actifs	13
1.2.2 Les capteurs passifs	13
1.3 Les approches mono-image	14
1.3.1 La reconstruction 3D à partir de l'intensité lumineuse	14
1.3.2 La reconstruction 3D à partir de la texture	15
1.3.3 La reconstruction 3D à partir du contour	15
1.3.4 Conclusion	17
1.4 Les approches multi-images	18
1.4.1 La reconstruction 3D par stéréovision	18
1.4.2 Combinaison de la stéréovision et du mouvement	21
1.4.3 Discussion	23
1.4.4 Reconstruction 3D à partir du mouvement	24
1.4.4.1 Les techniques discrètes	24
1.4.4.2 Les techniques continues	33
1.5 Que peut-on reconstruire sans calibrage?	36
1.6 Conclusion	37

2	Reconstruction 3D par vision dynamique	41
2.1	Introduction	41
2.2	Modélisation	42
2.3	Reconstruction 3D de primitives géométriques	44
2.3.1	Les points	47
2.3.2	Les droites	48
2.3.3	Les cercles	50
2.3.4	Les sphères	53
2.3.5	Les cylindres	55
2.3.5.1	Modélisation	55
2.3.5.2	Estimation des paramètres du cylindre à partir de deux limbes	58
2.3.5.3	Estimation des paramètres du cylindre à partir d'un seul limbe	61
2.3.6	Résultats de simulation	64
2.3.6.1	Exemple 1 : le point	64
2.3.6.2	Exemple 2 : la sphère	65
2.3.6.3	Exemple 3 : le cylindre	67
2.4	Conclusion	69
3	Reconstruction 3D par vision active	71
3.1	Introduction	71
3.2	Etat de l'art	74
3.2.1	Contrôle des paramètres oculaires	75
3.2.1.1	Le contrôle du focus	76
3.2.1.2	Le contrôle du diaphragme	76
3.2.1.3	La perception à champs variables	76
3.2.2	Tête binoculaire active	77
3.2.2.1	Le point de fixation	77

3.2.2.2	Le contrôle de la vergence	78
3.2.3	Estimation du mouvement 2D par vision active	79
3.2.4	Estimation du mouvement 3D par vision active	80
3.2.4.1	Cas d'un observateur monoculaire	81
3.2.4.2	Cas d'un observateur binoculaire	81
3.2.4.3	Discussion	82
3.2.5	Reconstruction 3D par vision active	82
3.3	Conclusion	86
3.4	Reconstruction 3D de primitives géométriques par vision active	87
3.4.1	Suppression des erreurs de discrétisation	88
3.4.2	Résultats de simulation	90
3.4.2.1	Exemple 1 : le point	90
3.4.2.2	Exemple 2 : la sphère	90
3.4.2.3	Exemple 3 : le cylindre	92
3.4.3	Minimisation des effets des erreurs de mesure	92
3.4.3.1	Cas du point	95
3.4.3.2	Autres primitives	96
3.4.4	Résultats de simulation	98
3.4.4.1	Exemple 1 : le point	98
3.4.4.2	Exemple 2 : la sphère	99
3.4.4.3	Exemple 3 : le cylindre	101
3.4.5	La commande référencée vision	103
3.4.5.1	Construction de la fonction de tâche correspondant à la reconstruction d'un point	105
3.4.5.2	Construction de la fonction de tâche correspondant à la reconstruction d'une droite	106
3.4.5.3	Construction de la fonction de tâche correspondant à la reconstruction d'un cylindre en utilisant un seul limbe	108

3.5	Conclusion	110
4	Appariement d'indices bidimensionnels	111
4.1	Introduction	111
4.2	Focalisation sur un segment de la scène	112
4.2.1	Introduction	112
4.2.2	Détection des points de contraste	113
4.2.3	Transformée de Hough	113
4.2.3.1	Définition	113
4.2.3.2	Mise en œuvre de la TH	114
4.2.4	Extraction des chaînes de points de contraste	115
4.3	Mise en correspondance de segments dans une séquence d'images . . .	116
4.3.1	Introduction	116
4.3.2	Choix des primitives	116
4.3.3	Méthodes de mise en correspondance	117
4.3.3.1	La corrélation	117
4.3.3.2	La programmation dynamique	118
4.3.3.3	La relaxation	118
4.3.3.4	Méthodes de prédiction-vérification d'hypothèses . .	119
4.3.3.5	Méthodes récursives	119
4.3.4	Conclusion	120
4.3.5	Détermination des appariements locaux de points de contour .	121
4.3.5.1	Modélisation d'un élément de contour en mouvement (ECM)	121
4.3.5.2	Test de vraisemblance pour la détermination des ECM :	123
4.3.5.3	Dérivation du critère sous forme d'une convolution :	124
4.3.5.4	Discussion	125
4.3.6	Application au suivi d'un seul segment	126
4.3.6.1	Calcul des ECM	126

4.3.6.2	Estimation des paramètres de la droite support des ECM	129
4.3.6.3	Rappel de l’algorithme	130
4.4	Présentation des résultats	130
4.4.1	Séquence “table”	131
4.4.2	Séquence “tuyaux”	133
4.4.3	Séquence “mire”	133
4.4.4	Implémentation temps-réel	137
4.5	Conclusion	138
5	Résultats expérimentaux	141
5.1	Description du banc expérimental	141
5.2	Reconstruction 3D d’un point	143
5.2.1	Reconstruction 3D d’un point par vision dynamique	144
5.2.2	Reconstruction 3D d’un point par vision active	146
5.2.2.1	Suppression de l’erreur de discrétisation	146
5.2.2.2	Minimisation des effets des erreurs de mesure	148
5.3	Reconstruction 3D d’une droite	151
5.3.1	Reconstruction 3D d’une droite par vision dynamique	151
5.3.2	Reconstruction 3D d’une droite par vision active	153
5.4	Reconstruction 3D d’un cylindre	158
5.4.1	Méthode de reconstruction 3D basée sur un seul limbe	158
5.4.1.1	Reconstruction d’un cylindre par vision dynamique	158
5.4.1.2	Reconstruction d’un cylindre par vision active	159
5.4.2	Méthode de reconstruction 3D basée sur deux limbes	168
5.4.2.1	Reconstruction 3D d’un cylindre par vision dynamique	168
5.4.2.2	Reconstruction 3D d’un cylindre par vision active	168
	Conclusion	177

A Les transformations géométriques	181
B Le calibrage	183
B.1 Introduction	183
B.2 Modèle de caméra	184
B.3 Distorsion de la lentille	185
B.4 Estimation des paramètres de la caméra	186
C Le système multiprocesseurs Sympati2	187
C.1 Introduction	187
C.2 Architecture de Sympati2	187
C.2.1 Arrangement hélicoïdal des données	188
C.2.2 Interconnexion entre les PEs	188
C.3 Environnement de programmation	190
C.3.1 Application au traitement d'images	190
Bibliographie	191

Introduction

La perception visuelle est certainement le sens le plus puissant parmi ceux dont l'homme est doté. Elle joue un rôle privilégié dans l'analyse du monde qui nous entoure. En effet, elle permet une interaction intelligente avec notre environnement ce qui explique le grand intérêt qui est porté aujourd'hui à la vision par ordinateur.

Depuis quelques années, de nombreux travaux menés en vision par ordinateur se sont fixés pour objectif la réalisation de systèmes capables d'accéder à la géométrie spatiale d'une scène à partir de son observation par une ou plusieurs caméras mobiles. Par exemple, pour maintenir l'autonomie des robots mobiles et leur permettre de réaliser des tâches de plus en plus complexes, il faut les doter d'outils de perception performants. En particulier, un module de reconstruction 3D robuste et fiable est souvent un maillon indispensable au système de perception du robot pour pouvoir explorer l'environnement dans lequel il évolue. Un tel module a pour but de fournir une description géométrique claire et complète de la scène à partir des images acquises.

Une telle compréhension du monde tridimensionnel, triviale pour l'homme, pose cependant de sérieuses difficultés en vision par ordinateur : les informations visuelles sont souvent bruitées, incomplètes voire instables et par conséquent difficilement exploitables.

De ce fait, il n'existe pas de systèmes réellement opérationnels en univers bruité et inconnu. Les études proposées aujourd'hui tentent d'apporter leur contribution au problème de la reconstruction d'environnements assez restreints (objets polyédriques

statiques, information a priori sur la scène, mouvements particuliers...) avec comme ambition la reconstruction de scènes de plus en plus complexes.

Les méthodes mises en œuvre sont généralement la stéréovision et la vision dynamique. Dans la première approche, un système de deux caméras, voire plus, est utilisé afin de remonter à une représentation tridimensionnelle de l'environnement perçu. La seconde approche utilise une caméra mobile qui, dans certains cas, est en plus commandable. On parle alors plus spécifiquement de vision active.

Comme son nom l'indique, la vision active ne se contente pas comme la vision dynamique d'*observer* l'environnement mais aussi de *explorer* et de *s'adapter* aux différentes situations qui se présentent en fonction de ce qu'elle perçoit. De manière générale, le but de la vision active est de manipuler les contraintes sous-jacentes aux phénomènes observés afin d'améliorer la qualité des résultats de perception [ALO87] [BAJ88] [SAN90] [BAN86].

Ce paradigme très récent a déjà montré à travers les quelques investigations qui sont apparues dans la littérature, qu'il régissait mieux les problèmes qui préoccupaient la communauté de la vision. Beaucoup d'algorithmes se sont montrés, à travers des études théoriques notamment [BAJ88], [ALO87], [BAN86], [XIE89], [BAL91], [BRO92], plus stables et plus robustes en utilisant la vision active. Ces travaux ont tous mis en valeur l'avenir très prometteur de la vision active dans tous les aspects de la vision et en particulier en reconstruction 3D, domaine dans lequel beaucoup de chercheurs s'acharnent encore à trouver des moyens qui permettent d'améliorer la robustesse des solutions.

Dans ce travail, nous proposons une contribution à la reconstruction de primitives géométriques paramétrables par vision active. Dans notre démarche, contrairement à la plupart des travaux en vision active qui se sont penchés sur la question, nous ne nous sommes pas arrêtés à une étude théorique souvent peu convaincante. Nous avons poussé l'investigation jusqu'à la validation finale en environnement réel.

Il est clair que l'étape la plus problématique dans la validation d'un travail, en robotique notamment, est incontestablement sa confrontation à une expérimentation réelle. En effet, toute application liée à la vision s'avère très capricieuse vis-à-vis des conditions d'éclairage, des distorsions du système de vision qu'on n'arrive que partiellement à modéliser (calibrage de caméras), de la complexité de la scène à interpréter, etc. Il ne faut bien sûr pas oublier, puisque c'est de vision dynamique

dont il s'agit avant tout, les imprécisions de la mesure des mouvements de la caméra qui sont les principales sources d'erreurs qui perturbent très notablement le processus de reconstruction 3D.

L'objectif principal de notre travail est de mettre en valeur l'apport très net de la vision active par rapport à la vision dynamique dans le domaine de la reconstruction 3D. Pour cela, nous avons examiné les deux points suivants :

- la modélisation, qui consiste à établir des méthodes de reconstruction 3D de primitives géométriques par vision dynamique, notamment pour des primitives non polyédriques.
- l'optimisation, qui consiste à améliorer par vision active la performance des méthodes mises en œuvre.

Par ailleurs, l'application qui nous intéresse tout particulièrement est la navigation et l'inspection à l'intérieur des bâtiments auxiliaires d'une centrale nucléaire. Les scènes réelles que nous devons donc interpréter sont essentiellement constituées de tuyaux que nous modélisons bien évidemment par des cylindres.

Pour que notre méthode puisse opérer en environnement réel, un algorithme de suivi des limbes des cylindres dans la séquence d'images perçue est indispensable. Nous proposons donc, pour compléter notre investigation dans la reconstruction d'objets cylindriques ou polyédriques, une méthode robuste à ce problème connu pour sa grande complexité.

Ce travail est structuré de la manière suivante :

- Dans le premier chapitre, nous présentons un panorama des différentes approches rencontrées dans la littérature en reconstruction 3D.
- Dans le second chapitre, nous présentons une méthode générale d'estimation des primitives géométriques paramétrables par vision dynamique. Nous l'illustrons ensuite par quelques exemples de primitives les plus représentatives : point, droite, cercle, sphère et cylindre. Des résultats de simulation complètent ce chapitre.
- Le chapitre suivant est consacré à la reconstruction 3D par vision active. Nous montrons que par des mouvements adéquats de la caméra, nous arrivons à

minimiser les effets des erreurs de mesure pouvant intervenir dans le processus d'estimation. Puis, nous décrivons un schéma de commande en boucle fermée qui permet de générer automatiquement ces mouvements particuliers de la caméra. Nous présentons dans ce chapitre aussi des résultats de simulation qui montrent l'apport de notre stratégie active de reconstruction 3D par rapport au cas précédent (vision dynamique).

- Le chapitre 4 est dédié aux aspects de traitements d'images concernant plus spécifiquement les objets cylindriques ou polyédriques. Une méthode de mise en correspondance de segments dans une séquence d'images est présentée. Cette méthode intègre un algorithme de calcul d'éléments de contours en mouvement. Nous présentons ensuite sa validation sur plusieurs séquences d'images ainsi que les performances obtenues lors de son implantation sur une machine parallèle. Notre approche de reconstruction 3D par vision active impose en effet des contraintes fortes en temps de calculs.
- Dans le dernier chapitre, nous présentons la validation de notre système de reconstruction 3D sur une scène similaire à une zone à l'intérieur d'une centrale nucléaire. Les résultats obtenus montrent les performances de notre approche.

Enfin, quelques perspectives sont tracées en guise de conclusion.

Chapitre Premier

La vision tridimensionnelle

Dans ce chapitre, nous décrivons les différentes méthodes existantes en matière de reconstruction 3D, tout en essayant de relever les avantages ainsi que les limitations de chacune d'elles. Nous insisterons tout particulièrement sur les méthodes basées sur l'utilisation d'une caméra en mouvement sur laquelle s'est porté notre choix.

1.1 Introduction

Les données tridimensionnelles d'une scène sont acquises en pratique soit directement, avec des capteurs mesurant la profondeur, soit indirectement, en exploitant par exemple les informations 2D issues d'une ou plusieurs caméras.

Il existe diverses méthodes en vision par ordinateur permettant de remonter à une représentation 3D de l'environnement perçu. On distingue principalement deux grandes classes d'approches qui sont elles-même partitionnées en plusieurs sous-classes différentes :

- (1) les approches mono-images : comme l'indique le nom qui leur est attribué, ces approches n'utilisent qu'une seule image pour remonter à l'information 3D d'une scène. Ces approches sont classées, suivant le type de primitives utilisées, en trois catégories distinctes [ALO88] :
 - la reconstruction 3D à partir de l'intensité lumineuse [MAR82], [PEN84], [FRA87], [HOR89b];
 - la reconstruction 3D à partir de la texture [OHT81], [KEN81], [WIT81], [ALO88];

- la reconstruction 3D à partir du contour [BRA84], [HAR88], [DHO89], [HOR89a].
- (2) les approches multi-images : contrairement aux approches précédentes, ces approches font appel à au moins deux images pour déterminer la structure 3D d'une scène. On distingue deux catégories d'approches qui diffèrent de par la structure même du système de vision utilisé :
- la reconstruction 3D par stéréovision qui utilise un système de deux, voire de trois caméras, [LUS87], [WEN92a], [ZHA92], [AYA89a].
 - la reconstruction 3D à partir du mouvement qui utilise une seule caméra mobile [AGG88], [HUA91], [VIA92], [CRO92], [WEL89], [ADI89]. Cette approche est aussi communément appelée la vision dynamique. Dans certains cas, la caméra est en plus commandable [BAN86], [XIE89], [SAN90]. On parle alors plus spécifiquement de vision active.

Enfin, notons que des contributions récentes combinant la stéréovision avec le mouvement ont été proposées [AYA89b], [ZHA90], [WEN92a], [OKU91]. Comme la plupart des approches *mixtes*, ces méthodes se sont avérées très intéressantes comme on pourra le voir au paragraphe qui leur sera consacré.

1.2 Acquisition de données tridimensionnelles

Deux types de capteurs sont utilisés en pratique pour acquérir des informations tridimensionnelles d'une scène :

- les capteurs actifs;
- les capteurs passifs.

Pour une application donnée, le choix du capteur adéquat dépendra du type même de la tâche à réaliser (navigation, inspection, etc), de la précision requise et bien évidemment du type d'environnement à explorer (scènes d'intérieur, environnement hostile, etc.).

1.2.1 Les capteurs actifs

Les capteurs actifs balayent la scène à observer par un faisceau énergétique et détectent en retour le faisceau réfléchi par les obstacles. Les exemples les plus répandus de tels systèmes sont le télémètre laser et la sonde à ultrasons qui sont utilisés surtout pour la navigation des robots mobiles.

Il convient de souligner ici la distinction à faire entre les termes “vision active” et “capteurs actifs”. En effet, la vision active désigne les systèmes de perception capables de manipuler les contraintes sous-jacentes aux phénomènes observés afin d’opérer dans des conditions optimales. Quant au capteur, il est dit actif lorsque celui-ci transmet de l’énergie. Il faut donc noter que la vision active comme la vision passive peuvent introduire aussi bien des capteurs actifs que passifs. Dans le cadre de notre travail, une méthode de reconstruction 3D par vision active en utilisant un capteur passif sera développée.

1.2.2 Les capteurs passifs

A la différence des capteurs précédents, ces capteurs ne transmettent aucune énergie. On distingue principalement les capteurs odométriques et, bien entendu, les systèmes de caméra. Les capteurs odométriques permettent d’obtenir directement une mesure approximative du mouvement qui peut ensuite être intégrée dans un processus de reconstruction 3D de l’environnement perçu. Les systèmes de caméras, fort utilisés dans de nombreuses applications en robotique, permettent de remonter à une représentation 3D de la scène perçue en utilisant le plus souvent des algorithmes de stéréovision ou de reconstruction 3D par vision dynamique. Deux images au moins sont en principe nécessaires pour contribuer à la compréhension du monde tridimensionnel en utilisant la vision. Cependant, si des informations supplémentaires sur la scène sont connues, on peut alors extraire des données 3D à partir d’une seule image.

Parmi les capteurs qui viennent d’être cités, le plus intéressant est sans nul doute la caméra vidéo. En effet, les télémètres laser sont coûteux et leur mise en œuvre est assez lourde. Quant aux capteurs à ultrasons, malgré leur faible coût, ils souffrent de leur faible précision. Par contre, la caméra est un capteur non seulement économe, peu encombrant et facile à mettre en œuvre sur un robot mobile mais il permet aussi d’aboutir, en faisant appel à des algorithmes appropriés de traitement d’images, à une description 3D très riche de l’environnement. Cette description est d’une grande

utilité dans de nombreuses applications en robotique telles que la reconnaissance et la localisation d'objets 3D, ou la navigation des robots mobiles.

En ce qui nous concerne, l'outil de perception 3D qui sera utilisé est une caméra embarquée sur un robot à six degrés de liberté. La commande des mouvements de la caméra permettra en outre de se focaliser facilement sur n'importe quelle région d'intérêt de l'environnement.

1.3 Les approches mono-image

1.3.1 La reconstruction 3D à partir de l'intensité lumineuse

La structure 3D d'un objet peut être déterminée en s'appuyant sur les variations de luminosité de l'image de cet objet. Ces variations dépendent non seulement de l'éclairage de la surface 3D mais aussi de la forme et de la nature de cette surface. Si les positions des sources lumineuses ainsi que les propriétés réfléchives de la surface observée sont connues, on peut alors extraire des informations tridimensionnelles sur l'objet observé [MAR82], [FRA87], [PEN84]. Plus précisément, le but est de retrouver la profondeur des points et/ou les normales en ces points à la surface de l'objet considéré. En principe, la carte des profondeurs suffit pour déterminer directement les normales à la surface. Mais, en pratique, ces profondeurs ne pouvant être estimées avec suffisamment de précision, il est souvent souhaitable de calculer les normales directement à partir de l'image. Les contraintes que l'on impose sur l'environnement sont principalement :

- la surface est lisse;
- les propriétés de réflectance sont uniformes sur toute la surface de l'objet, la surface étant souvent supposée Lambertienne¹;
- les sources lumineuses sont ponctuelles.

En se basant sur ces hypothèses, une fonctionnelle des paramètres de la surface est définie. En général, c'est une fonction non linéaire d'un grand nombre d'inconnues (profondeur et normale en chaque point) et il est souvent très difficile d'assurer

¹Une surface Lambertienne est une surface idéale qui diffuse parfaitement, elle rayonne avec la même luminosité dans toutes les directions.

la convergence du processus d'optimisation vers un extremum global. Pour établir cette fonctionnelle, on s'appuie sur une équation de contrainte liant la luminosité de l'image à l'orientation de la surface [HOR89b], [FRA87], [PEN84], la réflectivité étant connue pour toutes les orientations possibles d'une surface et l'observateur (caméra) étant supposé suffisamment éloigné de la scène.

Ces approches sont malheureusement très sensibles au bruit et leur mise en œuvre est délicate. De plus, les contraintes qu'elles introduisent pour assurer une solution unique sont rarement vérifiées dans le cas de scènes réelles.

1.3.2 La reconstruction 3D à partir de la texture

L'orientation des surfaces texturées peut être évaluée à partir d'un gradient de texture issu des variations de densité et de taille d'éléments de texture appelés *texels* [MAR82], [ALO88], [KEN81], [WIT81]. Les méthodes existantes imposent des contraintes sur la nature de la texture. On suppose en général que la géométrie du texel est connue. La forme peut alors être déduite à partir de la distorsion des objets perçus due à la transformation perspective. D'autres travaux considèrent que la texture est régulière, autrement dit, que les texels sont uniformément répartis sur les surfaces planes de la scène (un mur de briques par exemple). Cette régularité de la texture permet d'extraire des lignes parallèles qui sont exploitées afin de déterminer l'orientation des surfaces texturées [OHT81].

Le champ d'application de ce type d'algorithmes est très restreint. Les algorithmes existants sont souvent mis en échec devant des images réelles. D'autre part, il n'existe pas de moyen pour tester préalablement si une des hypothèses habituellement prises en compte pour résoudre le problème de la reconstruction 3D à partir de la texture (régularité, géométrie du texel,..) est valide pour la surface observée de la scène.

1.3.3 La reconstruction 3D à partir du contour

Un contour perçu peut correspondre à divers phénomènes physiques apparaissant dans la scène : les discontinuités de réflectivité (marque, ombre), d'orientation de surface (arête), de profondeur (occultation), de mouvement, etc (voir figure 1.1). Cependant, malgré la pertinence et la richesse de l'information relative à l'origine physique d'un contour d'une image, elle est souvent omise dans les processus d'analyse et d'interprétation d'images. En effet, on préfère généralement une analyse plus succincte du problème à une analyse qui serait pourtant plus élaborée avec l'intégra-

tion de l'information "type de contour" mais qui est en contre-partie plus complexe à obtenir.

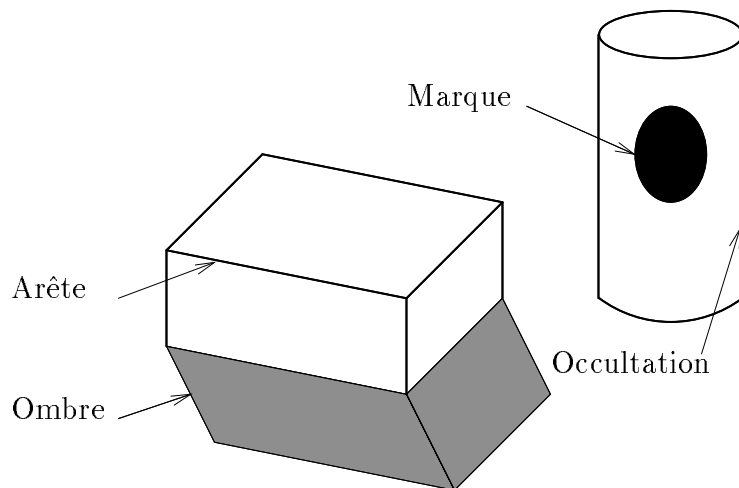


Figure 1.1 : *Différents types de contour*

Les approches qui déterminent la structure des objets à l'aide de leurs contours dans une image, [BRA84], [HAR88], [HOR89a], [DHO89], sont basées sur les propriétés géométriques de la projection perspective et aboutissent en général à des solutions analytiques complexes. Par exemple, la méthode proposée par [DHO89] est basée sur l'interprétation d'un triplet de droites 2D comme la projection perspective d'un triplet d'arêtes d'un modèle de l'objet considéré :

Pour localiser un objet dans le monde tridimensionnel, la mise en correspondance de trois droites de l'image de cet objet avec trois droites d'un modèle de cet objet s'impose. La méthode permet de calculer la transformation géométrique à appliquer au modèle de l'objet afin d'avoir une localisation consistante avec les trois droites choisies dans l'image. La détermination de cette transformation s'effectue en deux étapes : d'abord la rotation puis la translation. Un choix judicieux des deux repères image et objet permet de simplifier de manière très significative les équations à résoudre pour déterminer les paramètres de la rotation.

Les solutions (différentes localisations compatibles avec l'interprétation des trois droites) sont alors obtenues par la résolution d'une équation du 8ème degré. Lorsque les droites sont coplanaires ou concourantes le nombre de solutions est divisé par deux. Dans tous les cas, la bonne solution est toujours obtenue en considérant quelques règles simples. Notons que cette approche est étendue à certains types

de cylindres généralisés dont le contour image correspondant présente au moins un point d'inflexion [RIC91].

Le principal inconvénient de cette approche est qu'elle requiert un modèle précis de l'objet à localiser. De plus, cette méthode est sensible au bruit du fait du degré élevé des non linéarités qu'elle induit.

1.3.4 Conclusion

Toutes ces techniques de vision monoculaire font appel à des contraintes strictes d'éclairage ou de géométrie. Par ailleurs, ces techniques sont assez instables. De ce fait, elles ne peuvent constituer un système de vision 3D robuste et fiable pour un robot mobile. Cependant, elles pourraient être intégrées comme estimées initiales dans un système de vision dynamique par exemple.

Notons que ces trois approches distinctes de reconstruction 3D ont été reprises par Aloimonos *et al.* mais en utilisant la vision dynamique [ALO87]. Les auteurs ont montré que la redondance d'informations issues de plusieurs vues acquises par un observateur en mouvement permettait d'utiliser des contraintes plus générales et d'augmenter la robustesse et la stabilité des solutions :

- Dans le cas d'un observateur passif, le problème de la reconstruction 3D à partir de l'intensité lumineuse a une infinité de solutions et des contraintes supplémentaires sont indispensables pour garantir une solution unique. De plus, les algorithmes développés dans cette direction sont pour la plupart instables. Par contre, avec un observateur en mouvement (dynamique) le problème n'a plus qu'une solution qui se déduit à partir d'équations linéaires. La contrainte très couramment utilisée : "la surface de l'objet est lisse" devient inutile. En outre, cette solution est stable.
- Le problème de la reconstruction 3D à partir de la texture requiert, comme nous l'avons déjà souligné, des hypothèses sur la texture pour être résolu par un observateur passif. Par contre, un observateur dynamique est capable de recouvrer la structure sans hypothèse aucune et en utilisant des équations linéaires.
- Le problème de la reconstruction 3D à partir du contour est aussi un problème difficile. Dans le cas d'un observateur passif, des hypothèses doivent être po-

sées pour assurer une solution unique. Avec un observateur en mouvement, le problème a une solution unique obtenue en résolvant un système linéaire.

Nous allons à présent décrire les approches multi-images en commençant par la plus couramment utilisée : la stéréovision.

1.4 Les approches multi-images

1.4.1 La reconstruction 3D par stéréovision

La vision stéréoscopique permet de reconstruire la structure 3D d'une scène en l'observant depuis plusieurs points de vue différents avec un système de plusieurs caméras (en général deux ou trois). Cette technique a suscité depuis de nombreuses années un grand nombre de travaux de recherche, [LUS87], [ZHA90], [WEN92a], [ZHA92], [OKU91]. Pour une bonne revue des méthodes existantes le lecteur pourra se référer à l'ouvrage de Ayache [AYA89a].

La motivation initiale de cette classe de techniques était de simuler un mécanisme de perception utilisé par la plupart des êtres vivants : la triangulation, qui permet d'obtenir des coordonnées 3D à l'issue de la mise en correspondance des images de l'oeil droit et de l'oeil gauche.

La stéréovision présente trois principales caractéristiques :

- la notion d'appariement de primitives : pour reconstruire un objet à partir de ses projections, une mise en correspondance des primitives d'intérêt de cet objet dans les différentes images s'impose. La principale contrainte qui intervient dans ce processus est la contrainte épipolaire.
- la contrainte épipolaire : c'est la contrainte géométrique de base. Etant donné un point $m_1 = (X_1, Y_1)^T$ dans l'image 1, son correspondant $m_2 = (X_2, Y_2)^T$ dans l'image suivante se trouve nécessairement sur une droite D_{12} , appelée épipolaire du point m_1 (voir la figure 1.2). Le problème d'appariement se ramène alors à un problème monodimensionnel de recherche le long de la droite épipolaire. L'écart de position d entre les deux points image m_1 et m_2 correspondant à un même point $m = (x, y, z)^T$ de la scène est appelé disparité. Cette disparité s'exprime par :

$$d = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

- la triangulation : les disparités étant connues, on peut alors déterminer la profondeur des points 3D par triangulation après calibration du système stéréoscopique. Par exemple, si les deux caméras ont des axes optiques parallèles, les lignes épipolaires sont alors horizontales. Par conséquent, la disparité est définie par :

$$d = X_1 - X_2 = -f \frac{D}{z}$$

D étant la distance entre les deux centres optiques et f la distance focale. La disparité varie donc de manière inversement proportionnelle à la profondeur.

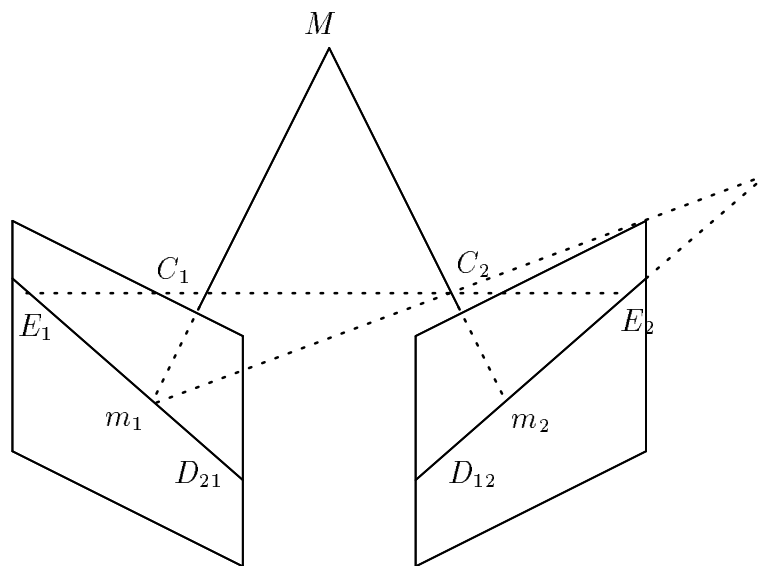


Figure 1.2 : Géométrie épipolaire

En plus de la contrainte épipolaire, les algorithmes de mise en correspondance existants s'appuient en général sur plusieurs autres contraintes afin de robustifier davantage le critère d'appariement retenu. Les plus utilisées sont les suivantes :

- la contrainte d'unicité qui spécifie qu'une primitive de la première image possède au plus un appariement dans la seconde image. Néanmoins, en pratique, des ambiguïtés sont souvent rencontrées.
- la contrainte de continuité qui, elle, spécifie qu'il n'y a pas de variation brutale de la profondeur des objets dans la scène, c.a.d qu'il n'y a pas de discontinuités importantes dans les surfaces observées. On peut alors utiliser comme critère

d'appariement le fait que les disparités de deux couples voisins dans les images doivent être proches.

La mise en correspondance de primitives bidimensionnelles est le problème majeur rencontré par les approches stéréoscopiques. De plus, intervient le problème d'occlusion qui est souvent source d'erreurs dans le processus d'appariement. Pour réduire ces effets, on peut toutefois réduire la distance entre les deux caméras mais en contre partie, l'erreur de localisation du point est augmentée.

La détermination de la structure d'une scène à partir de la stéréovision requiert bien évidemment la connaissance de la géométrie du capteur stéréoscopique. Les paramètres géométriques du capteur sont déterminés par une procédure de calibrage [TSA86], [TOS87], [CHA89] (voir l'annexe B pour un bref rappel sur la notion de calibrage ou d'étalonnage de caméras). Ces paramètres doivent être estimés avec une très bonne précision pour aboutir à des informations 3D exploitables en robotique.

Il faut noter cependant que même doté de deux "yeux" un système de vision stéréoscopique sera inévitablement confronté à des ambiguïtés dans un monde réel. On ne regarde pas seulement avec nos deux yeux mais aussi avec des mouvements permanents de ceux-ci et de la tête afin de lever les ambiguïtés.

Une manière de pallier cette absence de mouvement est d'utiliser une troisième caméra afin de permettre le renforcement de la contrainte épipolaire, ce qui permet de lever en partie ces ambiguïtés. En effet, on peut étendre de manière naturelle la méthodologie utilisée en stéréovision binoculaire au cas trinoculaire [LUS87], [AYA89a], [ZHA92]. L'apport d'une troisième caméra est très bénéfique dans la mesure où il simplifie les algorithmes de mise en correspondance et améliore la fiabilité du système. Cependant, le champ de vision commun devient encore plus restreint que dans le cas de deux caméras, c'est l'une des principales raisons qui peut dissuader d'adopter ce type de système malgré ses avantages.

La stéréovision telle que nous l'avons abordée jusqu'ici est connue sous le nom de *stéréovision passive*. On parle de *stéréovision active* lorsqu'un suréclairage de la scène est utilisé [GUI92], [WAN87]. Une des caméras est remplacée par une source de lumière structurée (par exemple un ensemble de points lumineux, une grille lumineuse,..). Ce dispositif permet de pallier les difficultés engendrées par l'étape fondamentale de mise en correspondance de primitives 2D. Le calcul de la

profondeur repose alors sur l'étude de la déformation de la structure de la lumière structurée perçue par la caméra.

1.4.2 Combinaison de la stéréovision et du mouvement

L'association ultérieure de la stéréovision avec d'autres systèmes d'acquisition d'informations tridimensionnelles permet d'obtenir des résultats très intéressants. En particulier, l'utilisation d'un système stéréoscopique dynamique s'avère plus efficace que les systèmes rigides précédents quant à la détermination de la structure et du mouvement 3D. En effet, le mouvement du capteur contribue de manière très significative à lever les ambiguïtés d'appariement qui apparaissent dans la scène. En outre, il permet, grâce à la redondance d'informations qu'il induit, d'améliorer la robustesse de l'estimation.

Quelques systèmes allant dans ce sens sont décrits ci-dessous. Tous combinent la stéréovision avec le mouvement. On passe ainsi d'une perception statique à une perception dynamique de l'environnement d'un robot [AYA89b], [WEN92a], [ZHA90], [OKU91].

Weng et al., [WEN92a], proposent une approche stéréoscopique pour déterminer le mouvement et la structure 3D. Les primitives de base sont des points 3D. Une longue séquence stéréoscopique est utilisée et une approche récursive est adoptée pour fusionner les multiples vues stéréoscopiques. Les problèmes de perception 3D sont en effet souvent formulés en terme de filtrage récursif. Le filtre de Kalman s'est révélé un outil puissant et général pour la résolution de ce problème. En effet, il permet de tenir compte de manière explicite de l'incertitude des données, d'exploiter leur redondance (beaucoup de systèmes fusionnant des informations fournies par des capteurs de types différents,...) et d'obtenir une représentation cohérente de l'environnement. La mise en œuvre du filtre de Kalman requiert néanmoins la donnée d'une estimée initiale du vecteur des paramètres à estimer ainsi que sa matrice de covariance.

Ayache et Faugeras dans [AYA89b], proposent une stratégie pour reconstruire de façon incrémentale l'environnement (statique) d'un robot mobile. Ils ne s'intéressent qu'à des scènes d'intérieur et les primitives de base sont des droites. Une paramétrisation minimale est utilisée pour chacune des deux entités (primitives et mouvement du capteur). Un système de vision trinoculaire [AYA89a] permet de reconstruire les droites 3D de l'environnement. Les descriptions locales sont alors fusionnées en des descriptions cohérentes et plus globales. Le modèle de l'environnement (incluant les paramètres de mouvement du robot) est alors mis à jour en utilisant un schéma

récuratif de prédiction-vérification intégrant un filtre de Kalman étendu.

Initialement, une estimée grossière du mouvement du robot entre les deux vues est connue. Pour chaque vue (triplet d'images) un ensemble de droites 3D est déterminé à l'aide de l'algorithme de stéréovision trinoculaire. L'estimée grossière du mouvement est appliquée à chaque droite 3D de la première vue puis la distance de Mahalanobis à toutes les droites de la seconde vue est calculée. La distance de Mahalanobis est la principale distance utilisée dans les processus d'estimation intégrant un filtre de Kalman. Elle utilise en effet toutes les informations fournies par le filtre à savoir, les paramètres estimés ainsi que leur incertitudes. Cette distance s'exprime par :

$$d_{\chi^2} = (\underline{p}^i - \underline{p}^j)^T (\Lambda_i + \Lambda_j)^{-1} (\underline{p}^i - \underline{p}^j) \quad (1.1)$$

où $\underline{p}^i, \underline{p}^j$ sont respectivement l'ensemble des paramètres caractérisant la première et la seconde primitives (des droites ici) et Λ_i, Λ_j les matrices de covariance associées.

Un test du χ^2 permet alors de décider de la validité de chaque appariement. Une fois la phase d'appariement effectuée, le filtre de Kalman étendu permet de calculer une nouvelle *meilleure* estimée du mouvement. Le processus est répété jusqu'à ce qu'il n'y ait plus de nouvel appariement enregistré.

Des résultats de bonne qualité obtenus sur des scènes réelles illustrent cette approche. Néanmoins, elle n'a pas été testée sur des scènes autres que des scènes d'intérieur qui paraissent relativement simples à analyser. En outre, elle suppose qu'une approximation du mouvement du robot est connue dès le départ.

Une autre étude, proche de celle qui vient d'être présentée sur l'analyse du mouvement à partir de vues stéréoscopiques, a été réalisée par Zhang [ZHA90]. Les primitives de base sont aussi des segments 3D reconstruits par stéréovision. L'auteur a mis au point deux algorithmes complémentaires d'estimation de la structure et du mouvement 3D de la scène perçue qui, contrairement aux deux cas précédents, est susceptible d'inclure plusieurs objets rigides en mouvement :

- un algorithme exploitant uniquement deux vues stéréoscopiques et correspondant à la phase initiale, où aucune information ni sur la structure ni sur le mouvement de la scène n'est connue. Cet algorithme est basé sur un schéma de prédiction-vérification d'hypothèses pour réaliser l'appariement des segments 3D et déterminer leur mouvement initial. La phase de génération d'hypothèses s'appuie essentiellement sur la contrainte de rigidité. Pour chaque hypothèse, une estimation du mouvement est obtenue puis propagée sur toute la scène.

L'hypothèse entraînant le maximum d'appariements est choisie comme le mouvement rigide entre les deux vues.

- un algorithme exploitant une longue séquence de vues stéréoscopiques. Dans cette seconde approche, chaque segment 3D est suivi d'une vue à l'autre et ses paramètres cinématiques sont estimés en utilisant un filtre de Kalman étendu. Les segments ayant des paramètres cinématiques similaires sont alors regroupés en objets rigides. La mesure de similarité utilisée est la distance de Mahalanobis qui est d'ailleurs aussi utilisée dans le processus de suivi des segments 3D. Cette approche permet ainsi de différencier les mouvements provenant d'objets multiples. Bien entendu, des objets différents ayant des mouvements proches ne pourront pas être différenciés et seront regroupés en un même objet.

L'avantage d'une telle approche est que la reconstruction des segments 3D devient plus fiable et plus précise grâce à l'intégration de vues stéréoscopiques multiples. En outre, elle a le mérite de n'utiliser aucune information a priori sur le mouvement ni sur la structure de la scène.

1.4.3 Discussion

La stéréovision semble être très bien adaptée à la perception globale de l'environnement. Cependant, elle n'a plus les mêmes performances lorsqu'il s'agit d'analyser une zone bien précise de l'environnement en vue d'une reconnaissance ou d'une manipulation en robotique. Cela est dû essentiellement à la structure rigide du système de stéréovision.

Par contre, l'utilisation d'une caméra mobile est beaucoup plus souple d'emploi et ne présente aucune limitation dans le mouvement. Par ailleurs, le traitement n'étant effectué que sur une seule image contrairement à un système de stéréovision, on peut donc accéder à des cadences de traitement plus élevées pour une configuration matérielle donnée.

Soulignons cependant que, comme nous l'avons vu au paragraphe précédent, les limitations de la stéréovision pouvaient être atténuées de manière notable en associant cette dernière à un autre procédé de perception, la vision dynamique en l'occurrence.

Nous allons maintenant présenter l'approche utilisant une caméra en mouvement que nous décrirons plus en détails que les méthodes précédentes puisque c'est celle que nous avons adoptée.

1.4.4 Reconstruction 3D à partir du mouvement

L'objectif de cette classe de techniques est de réaliser la reconstruction d'une scène à partir d'un certain nombre d'images acquises par une caméra embarquée sur un robot mobile parcourant un environnement statique ou éventuellement contenant des objets en mouvement.

On peut classer les approches traitant de ce problème en deux catégories principales :

- les techniques continues qui utilisent le champ dense 2D de vitesses apparentes [ADI85], [NEG87], [WAX87], [VER90], [ADI89];
- les techniques discrètes qui utilisent le déplacement dans les images de primitives particulières [AGG87],[CHI89], [BOU89a], [VIA92], [CRO92], [WEN92b], [WEL89], [WU 88], [MIT89], [THO93],

chacune d'elles étant elle-même divisée en deux sous-classes distinctes selon que le mouvement de la caméra est connu ou non. Dans ce dernier cas, la précision de la reconstruction est bien évidemment très étroitement liée à l'estimation du mouvement.

Dans ce qui suit, les travaux de reconstruction 3D qui seront présentés se restreindront pour la plupart, comme dans notre cas, à un environnement statique.

1.4.4.1 Les techniques discrètes

Ces approches sont dites *discrètes* car on raisonne en termes de déplacement discret (rotation R , translation T) et non pas en termes de vitesse (continue). Un mouvement relativement important est permis entre deux images successives.

Les techniques discrètes de reconstruction 3D par vision dynamique reposent sur trois étapes fondamentales :

- la première étape consiste à extraire dans la séquence d'images un ensemble de primitives 2D *suffisamment* pertinentes. Ces primitives sont en général des points caractéristiques, des segments de contour, des régions, des portions de courbe, des contours d'occultation, etc.
- l'étape suivante consiste à appairer les primitives sélectionnées d'image en image. On retrouve ainsi le même problème que celui de la stéréovision. Ce

problème extrêmement difficile fait encore l'objet de nombreux travaux au sein de la communauté scientifique. Des techniques diverses telles que la corrélation, la relaxation, la prédiction-vérification d'hypothèses,.. ont été étudiées. On reviendra de façon plus détaillée sur ce problème fondamental au chapitre 4.

- la dernière étape consiste, une fois la mise en correspondance des primitives 2D établie, à estimer le mouvement relatif caméra-scène² et la structure 3D de la scène. Des contraintes sont alors formulées sur la scène.

La prise en compte de déplacements importants entraîne naturellement une estimation plus robuste sur les paramètres de la structure 3D et du mouvement. Par contre, plus le mouvement sera important et plus la tâche de mise en correspondance des primitives 2D considérées sera délicate.

La principale contrainte prise en compte dans ce type d'approches est, comme dans le cas des approches stéréoscopiques, la contrainte de rigidité. Cette contrainte est utilisée soit de manière implicite : tous les points observables d'un même objet rigide subissent le même mouvement; soit de manière explicite : par exemple, la distance 3D entre deux primitives d'un même corps rigide est préservée durant le mouvement de la caméra. De telles contraintes engendrent en général un système d'équations non linéaires qui est résolu en exploitant entre autres les déplacements observés sur les primitives 2D sélectionnées.

Les techniques discrètes avec connaissance du mouvement

Crowley et al.,[CRO92], proposent une réponse au problème de la reconstruction 3D à partir du mouvement en supposant le déplacement de la caméra parfaitement connu. Leur primitives de base sont des droites. Une reconstruction incrémentale de l'environnement est réalisée en s'appuyant sur un schéma comprenant trois étapes essentielles : une étape de prédiction, une étape d'appariement et enfin la mise à jour du modèle du monde par un filtre de Kalman.

Les auteurs utilisent un processus de suivi de segments 2D³ perçus par la caméra au cours de son mouvement. Cette procédure qui s'appuie également sur un filtre de Kalman assure la maintenance d'une liste de segments de contour dynamiques. Ces derniers sont définis par un vecteur de paramètres 2D, les dérivées temporelles (vitesses apparentes) de ces paramètres ainsi que la matrice de covariance qui leur est associée.

²Cette estimation n'est pas toujours nécessaire, le mouvement de l'observateur pouvant être connu avec suffisamment de précision.

³Nous reviendrons plus en détails sur les problèmes de suivi de primitives 2D au chapitre 4.

Connaissant le déplacement de la caméra entre chaque image, les droites 3D sont alors reconstruites par triangulation. Plus précisément, les points extrémités des droites 3D sont reconstruits en utilisant les points 2D correspondants dans deux images de la séquence. Ces images sont choisies suffisamment espacées (5 images ici) afin d'obtenir une bonne précision de reconstruction. Ainsi, les segments 3D sont définis par une paire de points 3D ainsi que les incertitudes associées.

Une fois les segments 3D reconstruits, ils doivent être intégrés dans le modèle du monde. Pour cela, une mise en correspondance entre les segments 3D obtenus et ceux déjà existant dans le modèle s'impose. Cette procédure utilise la distance de Mahalanobis⁴. Les segments 3D se correspondant sont alors fusionnés pour mettre à jour les paramètres et les incertitudes des segments du modèle du monde.

Le problème de la reconstruction d'objets polyédriques a été relativement bien cerné ou du moins il a été largement abordé [FAU87], [XIE89], [BOU89a], [CRO92], [WEL89], [WEN92b], [AGG87]... Peu de travaux se sont penchés sur le problème plus général d'objets non polyédriques. Ces objets posent de nouveaux problèmes : les surfaces observées n'étant plus planes, les contours observés ne sont pas nécessairement rectilignes. Il se pose un problème de représentation : le choix de points ou de droites comme primitives, largement utilisés dans les méthodes précédentes, n'est plus envisageable. En outre, les objets non polyédriques (tridimensionnels) imposent une autre limitation plus fondamentale : les contours correspondant à différentes images de l'objet dépendent du point de vue. Ce sont des contours d'occultation qui sont caractérisés par le fait que les lignes de vue des points leur appartenant sont tangentes à la surface de l'objet considéré.

Dans [SAF92], les auteurs présentent une approche discrète pour la localisation de primitives telles que les cercles ou les ellipses, très fréquentes en robotique manufacturière (assemblage automatique de pièces mécaniques). La reconstruction de ces primitives non polyédriques s'avère évidemment plus complexe que celle concernant les primitives points ou droites.

Les auteurs présentent une méthode analytique qui consiste à décomposer le problème d'estimation en deux étapes. Ils déterminent d'abord l'orientation de la primitive; ensuite, ils en déduisent sa position. Connaissant le mouvement de la caméra entre deux images successives, le problème revient à déterminer l'intersec-

⁴Notons que pour cette procédure d'appariement les auteurs utilisent des paramètres redondants des segments 3D autres que les points extrémités qui s'avèrent évidemment peu fiables pour cette procédure.

tion entre deux cônes (un pour chaque position de la caméra) définis par leur base (projection perspective de la primitive dans le plan image) et leur sommet (centre optique correspondant à chaque vue). L'orientation de la primitive est déterminée en résolvant un système relativement complexe (du 4ème degré). Ayant déterminé le plan sur lequel se trouve la primitive, sa position est alors déterminée en calculant l'intersection du cône correspondant à la première vue avec le plan trouvé.

Nous montrerons dans le chapitre suivant qu'il est possible, notamment dans le cas de primitives circulaires et même sphériques ou cylindriques, de déterminer leur orientation simplement en résolvant un système linéaire. Notre approche est donc potentiellement plus robuste que celle de Safaei *et al.* En outre, notre formulation mathématique est beaucoup plus simple.

Les techniques discrètes sans connaissance du mouvement

Contrairement aux méthodes précédentes, ces approches, ne disposant pas d'informations suffisamment fiables sur le déplacement de la caméra, s'appuient sur un schéma d'estimation relativement plus complexe. Ce schéma consiste en effet à déterminer non seulement la structure 3D de la scène perçue mais aussi le mouvement de la caméra, ces deux problèmes étant intimement liés.

Pour illustrer ce problème, considérons le schéma classique suivant qui consiste à utiliser implicitement la contrainte de rigidité. Ce schéma conduit à l'équation de contrainte coplanaire qui lie les six paramètres du mouvement relatif caméra-scène décomposé en une translation T et une rotation R (voir l'annexe A qui traite de la représentation des mouvements rigides) et les projections d'un point de la scène dans deux images successives.

Soit un point $p = (x, y, z)^T$ de la scène qui se projette par projection perspective en un point $P = (X, Y)^T$ dans le plan image. Supposons que ce point se déplace dans la scène en un point $p' = (x', y', z')^T$ qui se projette en un point $P' = (X', Y')^T$ dans l'image. Si on suppose que le point p subit d'abord une translation T puis une rotation R alors on a [ALO90] :

$$p' = R(p + T) \Leftrightarrow R^T p' = p + T \Leftrightarrow T \wedge (R^T p') = T \wedge p \Leftrightarrow p(T \wedge (R^T p')) = 0.$$

En divisant la dernière équation par zz' on obtient l'équation suivante :

$$P(T \wedge (R^T P')) = 0 \tag{1.2}$$

appelée équation de contrainte coplanaire.

Le problème de la reconstruction à partir du mouvement est cependant limité à une ambiguïté échelle/vitesse : le mouvement rapide d'un observateur éloigné de la scène est indiscernable du mouvement lent d'un observateur proche de la scène. On peut doubler par exemple la valeur de la profondeur du point et l'amplitude de la vitesse de translation de la caméra tout en observant aucune différence dans l'image. Cette ambiguïté n'affecte que la translation, la rotation peut, elle, être entièrement déterminée. Ainsi, contrairement à la stéréovision, le problème de la reconstruction 3D par vision dynamique sans connaissance du mouvement de la caméra n'est résolu qu'à un facteur d'échelle près, le module de la translation ne pouvant être estimé. Le nombre de paramètres à déterminer est alors réduit à cinq. Il faut donc un minimum de cinq points pour déterminer le mouvement de la caméra à partir de la relation (1.2) si l'on n'a pas d'information sur le mouvement.

Parmi ces approches, il en existe qui utilisent deux ou trois images uniquement pour estimer la structure et le mouvement de la caméra [LIU86], [FAU87], [WEN89], [WEN92b], [WU 88], [MIT89], [THO93] d'autres exploitent une longue séquence d'images afin d'améliorer la robustesse de l'estimation [BOU89a], [VIA92].

Les observations introduites dans la formulation du problème sont en général des points 3D et leurs projections [FAU87], [WEN89], [BOU89a], [WU 88], [THO93] ou des droites 3D et leurs projections [LIU86], [FAU87], [WEN92b], [MIT89]. Dans ce dernier cas, deux images sont insuffisantes pour déterminer le mouvement de la caméra. Un minimum de trois images est en effet requis. Cela provient du fait que les droites 3D possèdent un degré de liberté supplémentaire par rapport aux points 3D.

L'utilisation de droites comme observations semble plus avantageuse que l'utilisation de points. En effet, l'extraction des droites est plus robuste au bruit. En outre, il est plus aisé de mettre en correspondance des droites que des points.

Parmi les algorithmes qui ont été développés pour déterminer la structure et le mouvement de la caméra, on peut discerner deux catégories :

- les algorithmes non linéaires qui exploitent directement les équations non linéaires liant le mouvement de la caméra au mouvement apparent 2D. Ces algorithmes sont coûteux en temps de calcul. En outre, ils peuvent aboutir à des solutions fausses (minima locaux). Il convient donc de posséder une bonne approximation de la solution recherchée pour initialiser le processus d'optimisation.

- les algorithmes linéaires qui calculent dans un premier temps un ensemble de paramètres intermédiaires en résolvant des équations linéaires. Ensuite, les paramètres de mouvement sont déduits de ces paramètres intermédiaires. L'avantage de ces algorithmes par rapport aux précédents réside dans le fait qu'ils sont rapides et garantissent généralement une solution unique. Néanmoins, des cas dégénérés existent et mettent en échec ces approches. En général, la solution d'un algorithme linéaire est sous-optimale. Cela est principalement dû aux différentes approximations qui sont faites. Par contre, elle peut constituer une excellente solution initiale pour un algorithme d'optimisation non linéaire.

Nous présentons ci-dessous une liste non exhaustive de travaux qui ont été développés dans les deux directions. Le lecteur intéressé pourra se référer à la synthèse bibliographique (peu récente certes) réalisée à ce sujet par Aggarwal [AGG88].

Weng et al. dans [WEN89] proposent une méthode linéaire de calcul du mouvement et de la structure 3D d'une scène à partir de la mise en correspondance de points sur deux images. L'algorithme requiert un minimum de huit points mais il en exploite un nombre plus important afin d'améliorer la robustesse au bruit. La structure 3D des points est alors estimée par moindres carrés.

La structure 3D étant reconstruite directement à partir des points observés, elle est sujette à des bruits de mesure. Une correction est alors effectuée en s'appuyant sur la contrainte de rigidité. Les auteurs tentent aussi d'estimer l'erreur commise sur les paramètres du mouvement de la caméra. Leur modélisation est applicable à toute méthode faisant appel à une technique de moindres carrés. Par ailleurs, une bonne analyse est présentée concernant les effets des perturbations sur, d'une part, les paramètres de mouvement et, d'autre part, la structure de la scène. Pour un même niveau de bruit, les erreurs ne sont pas toujours du même ordre pour différentes configurations spatiales de points et différents mouvements de la caméra. Les auteurs montrent entre autres que :

- le champ de vue de la caméra doit être large pour obtenir de meilleures estimées.
- l'amplitude de la translation doit être importante pour permettre une estimation stable de la direction de la translation et de la structure.
- une translation parallèle au plan image assure une estimation plus stable.
- la scène doit être proche de la caméra.

- les paramètres de rotation n'ont pas d'influence significative sur la fiabilité des solutions.

Liu et Huang, [LIU86], ont développé un algorithme pour l'estimation du mouvement de la caméra à partir de la mise en correspondance de droites 3D d'une scène rigide. Les principes de base sont les mêmes que les algorithmes linéaires basés sur les points 3D.

Les auteurs décomposent le mouvement de la caméra en une rotation suivie d'une translation. Ils montrent que dans le cas d'une rotation pure, la mise en correspondance de deux droites sur deux images est suffisante pour déterminer la rotation. Le système non linéaire obtenu est résolu par une méthode itérative. Dans le cas d'une translation pure, la mise en correspondance de cinq droites sur trois images conduit à un système linéaire donnant directement les paramètres de la translation. Dans le cas général, ils utilisent six droites sur trois images. Le principal inconvénient de cette méthode réside dans le fait que les paramètres de la translation sont fortement dépendants de la qualité de l'estimation des paramètres de la rotation.

Une approche plus robuste qui intègre de manière explicite les incertitudes sur les mesures est celle présentée dans [FAU87]. Cette méthode utilise un filtre de Kalman étendu afin d'obtenir une "meilleure" estimation des paramètres du mouvement de la caméra. La structure des droites 3D est alors déterminée en utilisant soit une technique de moindres carrés, soit un filtre de Kalman.

Une approche similaire a été développée par Boukkari [BOU89a] mais en utilisant des primitives points correspondant à des jonctions dans la scène perçue. Le déplacement de la caméra est supposé imparfaitement connu et la profondeur moyenne de la scène à la caméra est initialement connue.

Dans [WEN92b], les auteurs proposent aussi une méthode linéaire pour l'estimation du mouvement et de la structure 3D basée sur l'utilisation de droites. Trois images sont utilisées et un minimum de treize droites est requis dans cette formulation. Cette approche garantit une solution unique pour les paramètres de mouvement en dehors de certains cas dégénérés. Les auteurs présentent quelques conditions nécessaires et suffisantes sur les configurations de droites conduisant à une matrice de coefficients dégénérée. En présence de bruit, cette méthode linéaire n'est pas suffisamment robuste. Ils proposent donc une optimisation non linéaire initialisée par leur algorithme linéaire. Cette approche consiste à minimiser les erreurs observées dans les images par rétro-projection. Plus précisément, on minimise l'écart entre les valeurs estimées des paramètres des droites (calculés par moindres carrés) et les va-

leurs induites par les paramètres de mouvement 3D estimés. Cette fonction intègre les matrices de covariance associées aux droites considérées et peut être facilement étendue à d'autres primitives que les droites. La minimisation d'une telle fonction non linéaire est réalisée par un processus itératif.

Cet algorithme a le mérite d'exploiter la redondance des données afin d'améliorer la robustesse au bruit. Néanmoins, il paraît difficile à mettre en œuvre en pratique vu le nombre important d'appariements de droites qu'il requiert. En effet, apparier de manière fiable treize triplets de droites est loin d'être une tâche aisée. Encore faudrait-il que treize droites suffisamment significatives (longueurs supérieures à un seuil) puissent être détectées dans la scène.

Enfin, notons qu'il existe des approches qui combinent les deux types de primitives (points et droites). Dans ce cas, quatre points et une droite sur deux images sont suffisantes pour calculer la structure et le mouvement 3D [AGG87].

Cas d'objets non polyédriques

Une contribution récente concernant la reconstruction 3D de surfaces gauches a été réalisée par Arbogast dans son travail de thèse [ARB91]. L'auteur propose tout d'abord une méthode robuste de segmentation des contours d'occultation correspondant à ces surfaces gauches par des B-splines. L'observation continue de ces contours d'occultation sur une longue séquence d'images permet d'engendrer des surfaces spatio-temporelles en supposant bien entendu qu'une mise en correspondance entre contours soit établie⁵. Cette surface spatio-temporelle est paramétrée en faisant appel à la géométrie épipolaire.

L'auteur utilise un modèle de projection sphérique de la caméra qui permet d'aborder le problème de manière plus souple. Il propose alors une méthode de reconstruction de surfaces gauches en s'appuyant sur des propriétés différentielles des surfaces et des contours. Une solution est présentée concernant l'estimation du mouvement de la caméra à partir de l'observation de contours d'occultation des objets. Les courbes gauches observées sont supposées être rigides et fixes. Les paramètres de mouvement de la caméra sont déterminés à l'aide d'une méthode simple de résolution d'équations différentielles.

Des expérimentations réelles montrent la faisabilité de l'approche. Bien que cette méthode soit attrayante du point de vue théorique, elle présente l'inconvénient d'être

⁵L'auteur n'a pas approfondi cette question, cette phase de mise en correspondance étant faite manuellement.

très sensible au bruit du fait de son aspect différentiel. Par ailleurs, cette méthode ne permet pas de reconstruire les surfaces ne présentant aucun contour d'occultation (les surfaces planes par exemple). Notons qu'une approche similaire a été développée par Vaillant [VAI90].

Viala dans [VIA92] propose une méthode discrète pour la reconstruction 3D d'une scène cylindrique (contenant plusieurs cylindres). La méthode consiste à reconstruire de manière récursive les cylindres de la scène tout en effectuant un recalage de la caméra. Elle est mise en œuvre par un filtre de Kalman qui dispose bien sûr d'une connaissance grossière du déplacement de la caméra. Ainsi, le filtre estime *simultanément* les paramètres des différents cylindres présents dans la scène ainsi que les paramètres de position-orientation de la caméra. Néanmoins, pour que cette estimation soit possible, au minimum quatre cylindres (en supposant qu'un seul limbe des cylindres est observé) et sept images sont nécessaires compte tenu du nombre important de paramètres à estimer.

La reconstruction d'un cylindre s'appuie sur l'observation d'un seul de ses limbes⁶. Cependant, pour améliorer les performances du filtre, un appariement des limbes gauche et droit a été introduit dans la boucle d'estimation. Cet appariement repose sur l'utilisation de la distance de Mahalanobis. Cette distance utilise toute l'information fournie par le filtre de Kalman sur les paramètres des cylindres : les valeurs estimées et leur incertitudes. Ce module d'appariement n'est déclenché qu'après un certain nombre d'itérations afin de s'assurer que le filtre est en cours de convergence.

Cette méthode a été validée avec succès sur une scène contenant cinq cylindres. La principale limitation de cette approche réside dans le fait qu'elle impose que la scène perçue contienne au moins quatre cylindres ce qui n'est pas toujours possible dans un environnement réel. Notons cependant que la présence de seulement deux cylindres est indispensable lorsque les deux limbes de chaque cylindre sont observables et ont été correctement appariés.

Nous proposons nous aussi une contribution à la reconstruction de primitives cylindriques au chapitre suivant. Notre approche se distinguera de ces méthodes classiques, comme on le verra, par le fait que les informations sur les primitives d'intérêt de la scène ne sont pas acquises en aveugle mais de manière très sélective.

⁶les spécificités relatives aux primitives cylindriques sont décrites dans le chapitre suivant où nous abordons également la reconstruction de ce type de primitives.

Nous allons à présent brièvement évoquer les techniques continues. Ces méthodes, bien que moins utilisées que les précédentes, ne manquent pas d'intérêt. En particulier, elles ne requièrent pas de phase de mise en correspondance, souvent problématique, comme le cas des approches précédentes.

1.4.4.2 Les techniques continues

Ces techniques s'appuient sur une notion fondamentale : le mouvement apparent 2D. Pour déterminer ce champ de vitesses, une approximation au premier ordre, [WAX87], [BOU87], [BOU89b], [SCH89], [SUB89], est introduite, conduisant à l'équation de base reliant le vecteur vitesse dans l'image $\underline{\dot{X}} = \left(\frac{dX}{dT} \right)$ au gradient

spatial de la fonction intensité f , $\nabla f = \left(\frac{\partial f}{\partial X}, \frac{\partial f}{\partial Y} \right)^T$ et la dérivée temporelle de f , $\frac{\partial f}{\partial t}$:

$$\underline{\dot{X}} \nabla f = -\frac{\partial f}{\partial t} \quad (1.3)$$

Notons que cette équation suppose que l'intensité des points dont on cherche la vitesse apparente est invariante dans l'intervalle de temps considéré (entre deux images successives).

Contrairement aux approches discrètes, ces méthodes ne conviennent que pour de faibles déplacements de la caméra. Il est important donc de souligner ici la différence fondamentale entre ces deux classes d'approches :

- les approches discrètes s'appuient sur une formulation en termes de *déplacements* :

$$\underline{p}_{t+1} = R \underline{p}_t + T$$

R et T étant respectivement la matrice de rotation et le vecteur de translation de la caméra entre l'instant t et l'instant $t+1$.

- les approches continues s'appuient, elles, sur une formulation en termes de *vitesses* :

$$V(p) = \underline{\dot{x}} = -T - \Omega \wedge \underline{x}$$

où Ω et T sont respectivement la composante de rotation et la composante de translation du torseur cinématique de la caméra et V la vitesse d'un point 3D p de coordonnées $\underline{x} = (x, y, z)^T$ dans le repère caméra.

Les premières approches s'appuient, comme nous l'avons déjà vu, sur la mise en correspondance de primitives éparses extraites de chaque image. Les secondes se basent, elles, sur la détermination d'un champ dense 2D de vitesses apparentes. Tout comme le problème de la mise en correspondance de primitives 2D dans deux images, la détermination de ce champ reste un problème ouvert à ce jour, compte tenu des difficultés, notamment en présence de perturbations, qu'ont les approches existantes pour l'estimer convenablement.

Notons qu'il existe des approches hybrides, [ESP87], [XIE89], qui s'appuient sur une formulation en termes de vitesses dans le processus de reconstruction 3D tout en utilisant une mise en correspondance "discrète" de primitives éparses (pas de recours donc au gradient spatio-temporel de l'intensité).

Une fois que le mouvement apparent a été déterminé, des contraintes supplémentaires (contrainte de rigidité) concernant la scène sont considérées afin de pouvoir calculer la structure et le mouvement 3D. Une formulation mathématique à ce problème peut être trouvée dans [AGG88], [SUB87].

Comme les approches discrètes, les approches continues peuvent être aussi classées en deux catégories distinctes :

- les techniques continues sans connaissance du mouvement de la caméra;
- les techniques continues avec connaissance du mouvement de la caméra.

Les techniques continues avec connaissance du mouvement de la caméra

Dans [VER90], est proposée une approche pour la détermination de la structure à partir du mouvement intégrant une information clef : la trajectoire de la caméra est prédéfinie. Deux types de trajectoires sont considérés : une constituée d'une translation le long de l'axe optique, l'autre est une rotation autour d'un point fixe. Le mouvement de la caméra ainsi contraint simplifie énormément le calcul du mouvement apparent.

Une fois le mouvement apparent déterminé, la carte des profondeurs est calculée en utilisant les paramètres de la trajectoire de la caméra. L'application visée par ce travail concerne la saisie d'objets en vrac par un bras manipulateur. Des expérimentations réelles ont confirmé la faisabilité de ce système quant à une manipulation effective dans l'industrie.

Xie, [XIE89], utilise une caméra dont le mouvement est contrôlable et supposé connu. Il montre qu'il existe un mouvement 2D apparent optimal qui permet de minimiser l'erreur de reconstruction 3D et l'obtention du maximum d'informations sur la scène. Ce mouvement est tel que le champ de déplacements perçu dans le plan image soit dans une direction perpendiculaire à un sous-ensemble le plus grand possible des segments 2D estimés. Le mouvement local de la caméra assurant ce mouvement 2D optimal est généré par une boucle de contrôle rudimentaire intégrée dans le système de vision.

Cette approche se limite au cas d'objets polyédriques, les primitives de base étant des segments. La méthode consiste d'abord à mettre en correspondance les segments 2D dynamiques en s'appuyant sur un algorithme de calcul d'éléments de contours en mouvement [BOU89b]. Une formulation simple et générale permet alors de reconstruire les segments 3D. Enfin, une technique de moindres carrés fusionnant les segments 3D estimés en différents instants permet d'améliorer la précision de la reconstruction.

Cette approche est ainsi à cheval sur les deux techniques continue et discrète. Elle est continue dans la phase de reconstruction 3D mais, par contre, discrète quant à l'étape d'appariement de primitives 2D. Une approche similaire mais travaillant sur les points est décrite dans [ESP87].

Les techniques continues sans connaissance du mouvement de la caméra

Comme dans le cas des approches discrètes (sans connaissance du mouvement), l'estimation du mouvement de l'observateur, quand il ne peut être mesuré précisément, est une étape indispensable pour déterminer la structure 3D de la scène perçue.

Dans [NEG87], est présentée une méthode permettant de déterminer l'orientation d'une surface plane et le mouvement 3D de la caméra. Cette méthode n'utilise pas explicitement le mouvement apparent mais les dérivées spatiales et temporelles de l'intensité lumineuse. Un schéma itératif est introduit pour résoudre le système non linéaire qui découle d'une formulation du problème par moindres carrés. Des résultats, obtenus sur des images synthétiques, montrent que cet algorithme converge assez rapidement vers la bonne solution.

Adiv, [ADI85], traite le cas relativement plus complexe d'une caméra en mouvement dans un environnement contenant plusieurs objets mobiles. L'approche consiste

en deux étapes fondamentales :

Dans la première phase, le mouvement apparent est segmenté en régions élémentaires; chaque région étant cohérente avec le mouvement rigide d'une surface plane. Ce regroupement est réalisé à l'aide d'une transformée de Hough généralisée⁷. L'ensemble des vitesses apparentes appartenant à une même région est alors utilisé pour calculer les paramètres de mouvement correspondants, optimaux au sens des moindres carrés. La prochaine phase consiste à regrouper les régions dont les mouvements 3D correspondants sont proches. On suppose encore, en s'appuyant sur la contrainte de rigidité, que de tels groupes de régions correspondent au mouvement d'un même objet rigide. Il persiste bien évidemment une ambiguïté quant à la décomposition de la scène en objets mobiles indépendants. En effet, deux objets évoluant indépendamment peuvent induire, dans certains cas, des mouvements 2D affines pouvant être interprétés comme résultant du mouvement d'un même objet rigide.

Une fois que le champ des vitesses apparentes a été segmenté, le but est de déterminer les paramètres du mouvement 3D relatif caméra-objet et les profondeurs relatives des points correspondants dans chaque région. Pour cela, une technique de moindre carrés est utilisée. Elle consiste à minimiser l'écart entre les valeurs mesurées du mouvement 2D et celles prédites à partir de la structure et du mouvement 3D estimés.

1.5 Que peut-on reconstruire sans calibrage?

Les systèmes de vision 3D que nous venons de présenter requièrent tous une phase préliminaire indispensable : le calibrage. Mohr et al., [MOH91], [MOR93], [FAU92], montrent que l'on peut apporter des solutions aux problèmes rencontrés par la vision 3D en s'affranchissant d'une phase de calibrage *directe*. Pour cela, ils s'appuient sur des constructions géométriques simples issues de propriétés de la géométrie projective. Ils considèrent en effet les propriétés invariantes par projection perspective.

Le principal outil mathématique utilisé est le birapport qui a la propriété d'être invariant par projection perspective. En se basant sur cet outil, ils montrent que :

- on peut reconstruire à partir d'une seule vue perspective la ligne de vue associée à un point de l'image et ce, sans connaître la position de la caméra dans la scène. Il est aussi possible de reconstruire la position du centre optique à partir d'une seule vue.

⁷On trouvera un rappel des principes de la transformée de Hough dans le chapitre 4.

- en stéréovision, on peut déterminer la droite épipolaire associée à un point m de l'une des deux images. On apporte ainsi une réponse au problème fondamental de la stéréovision.

Ces deux problèmes sont résolus à l'aide de deux groupes de quatre points de référence coplanaires. La même technique est applicable en utilisant six points de référence non coplanaires. Les auteurs montrent aussi comment localiser, en utilisant cette approche, des objets relativement les uns aux autres.

Cette méthode très simple à mettre en œuvre peut être très avantageuse dans certaines situations. Si la scène contient des balises dont la position 3D est connue (points de référence), l'étape fastidieuse de calibrage de la caméra peut être totalement ignorée et les paramètres nécessaires à une manipulation en vision tridimensionnelle peuvent être directement obtenus par la méthode qui vient d'être décrite. D'autre part, si la position de ces balises n'est pas connue, un positionnement relatif à ces balises est possible. Il suffit tout simplement de leur affecter des positions 3D arbitraires.

Malgré l'apport nouveau de ce types d'approches aux problèmes de vision tridimensionnelle, elles restent quand même difficilement envisageables dans des conditions réelles de manipulation (environnement inconnu, hostile,..). De plus, ces méthodes nécessitent une très grande précision sur les mesures dans l'image.

1.6 Conclusion

Nous venons de présenter dans ce chapitre les principales approches permettant de déterminer le mouvement et la structure 3D d'une scène. Les approches utilisant une seule image de la scène se sont avérées peu fiables et sensibles au bruit. Leur champ d'application est restreint. Néanmoins, elles peuvent constituer une bonne solution initiale dans un processus de reconstruction 3D faisant appel à une séquence d'images. Les approches stéréoscopiques quant à elles se sont avérées beaucoup plus prometteuses tant en précision qu'en fiabilité. Cependant, leur utilisation en environnement réel est relativement lourde, en particulier quand il s'agit d'inspecter une zone bien précise de l'environnement. La vision dynamique nous paraît beaucoup plus souple d'utilisation tout en ayant des performances similaires à celles des approches stéréoscopiques.

Nous avons déjà mis l'accent sur les principaux inconvénients de chacune des

deux classes de techniques (discrète et continue) de vision dynamique. Nous les rappelons encore ici :

- La principale difficulté rencontrée par les approches discrètes est la phase de mise en correspondance d'indices bidimensionnels entre deux images successives. Cette mise en correspondance est souvent délicate et implique l'emploi d'algorithmes complexes et coûteux en temps de calcul. En outre, aucune stratégie de déplacement de la caméra n'est utilisée. Le robot agit en aveugle et n'exploite pas toutes les informations qu'il pourrait acquérir au cours de son déplacement.
- L'inconvénient majeur des approches continues est leur sensibilité au bruit due au fait qu'elles imposent l'utilisation de faibles déplacements entre les images successives. En outre, des discontinuités peuvent apparaître dans les champs de vitesses et perturber notablement l'estimation si elles ne sont pas détectées de manière fiable.

Nous avons opté pour une approche hybride combinant les deux méthodes précédentes. Notre approche, comme les approches continues, s'appuie sur une formulation en termes de vitesses dans la phase de reconstruction 3D. Par contre, elle utilise, comme les approches discrètes, la mise en correspondance dans les images de primitives éparses. On s'affranchit ainsi de l'étape problématique de détermination du champ 2D de vitesses apparentes. En outre, du fait de son aspect continu, la mise en correspondance des primitives 2D est largement simplifiée. Il est très important d'assurer de manière fiable le suivi des primitives dans la séquence d'images. La mise en échec du suivi, qui n'est pas à négliger vu le nombre d'ambiguïtés qui peuvent être rencontrées dans un monde réel, peut engendrer des résultats de reconstruction catastrophiques.

D'un autre côté, la prise en compte de faibles déplacements de la caméra conduit à une reconstruction 3D de la structure moins robuste que celle obtenue par les approches discrètes classiques. Pour pallier cet inconvénient, nous proposons une approche en boucle fermée qui va permettre justement de compenser toutes les erreurs de mesure qui perturbent le processus de reconstruction. Nous verrons qu'il sera même possible, malgré l'aspect continu de notre approche, de prendre en compte des déplacements importants entre les deux (ou plus) images prises en compte dans l'estimation de la structure 3D.

Dans la suite de ce travail, nous présenterons d'abord un formalisme élégant permettant de reconstruire toute primitive géométrique paramétrable. Ensuite, nous utiliserons une loi de commande en boucle fermée intégrant directement les informations visuelles pour générer de manière automatique les mouvements de la caméra. Ces mouvements sont particuliers : ils permettent une reconstruction 3D optimale. Ce processus de reconstruction adaptatif permet d'améliorer constamment l'estimation de la structure 3D au cours du temps. C'est une approche de type *vision active*.

Chapitre 2

Reconstruction 3D par vision dynamique

Dans ce chapitre, nous proposons une méthode générale de reconstruction 3D des primitives géométriques paramétrables par vision dynamique. Cette méthode est basée sur un concept fondamental : le torseur d'interaction associé aux primitives de l'environnement qui caractérise entièrement les interactions entre les mouvements de la caméra et la primitive considérée. Cette méthode est appliquée aux primitives les plus représentatives : point, droite, cercle, sphère et cylindre.

2.1 Introduction

Comme nous venons de le voir dans le chapitre précédent, le problème de la structure à partir du mouvement est un problème très complexe qui reste ouvert, notamment dans le domaine des techniques continues.

En ce qui nous concerne, nous désirons construire une représentation géométrique de l'environnement à partir des images obtenues par une caméra embarquée sur l'effecteur d'un robot. Pour cela, nous supposons que le mouvement du robot est parfaitement connu. Nous n'aurons donc pas à estimer les paramètres du mouvement de la caméra. Le problème reste quand même très difficile comme en témoignent les travaux que nous avons présentés au chapitre précédent et qui reposaient sur la même hypothèse [XIE89], [CRO92], [SAF92], [VER90]. Par ailleurs, nous avons déjà souligné le fait que l'estimation de la structure était intimement liée à celle du mouvement et que le principal souci des deux approches (avec ou sans connaissance du mouvement) était de faire face au bruit qui perturbe notablement les estimations.

Comme la quasi-totalité des travaux développés dans le domaine de la structure à partir du mouvement, nous restreindrons notre étude à des scènes statiques. Déterminer correctement la structure d'objets statiques mêmes les plus simples tels que les objets polyédriques reste encore une tâche très problématique [XIE89], [BOU89a], [CRO92], [WEL89], [WEN92b]. Il est relativement aisé de trouver des méthodes analytiques permettant de remonter à la structure 3D des objets. Cependant, en dépit de toutes les restrictions que l'on impose sur l'environnement (éclairage, objets opaques, bien contrastés, entièrement visibles,...) les résultats en pratique restent très en dessous des exigences des applications industrielles. Par ailleurs, ces méthodes manquent de généralité et ne sont dédiées souvent qu'à une catégorie très restreinte de primitives de l'environnement [BOU89a], [XIE89], [CRO92], [SAF92], [VIA92].

Notre contribution consiste justement en une tentative d'aborder le problème de la reconstruction 3D de primitives de manière plus générale [BOU92b]. Contrairement donc à la plupart des travaux qui se sont consacrés uniquement à des primitives point [BOU89a], droite [XIE89], [CRO92], [WEL89], [WEN92b], droite et cylindre [VIA92], cercle [SAF92],... nous présentons une méthode générale adaptable à n'importe quelle primitive géométrique paramétrable et en particulier à chacune des primitives qui viennent d'être citées pour lesquelles nous proposons une solution explicite [BOU92b], [CHA92], [BOU93c].

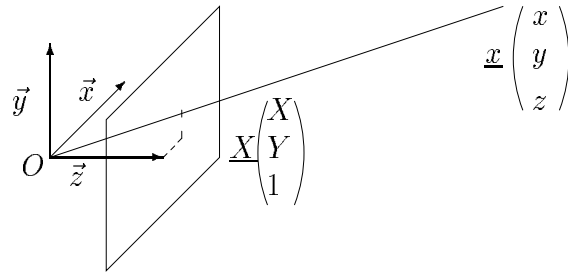
Ainsi, on pourra reconstruire un champ relativement vaste de scènes constituées d'objets dont la structure spatiale peut être approximée par des ensembles de primitives géométriques paramétrables (droites, cylindres...).

2.2 Modélisation

Signalons tout d'abord que le repère de la caméra $(O, \vec{x}, \vec{y}, \vec{z})$ donné Figure 2.1 sera par défaut celui dans lequel seront exprimées par la suite toutes les grandeurs nécessaires, par exemple, le torseur cinématique, les coordonnées des points, les torseurs d'interaction, etc.

La caméra est modélisée par son centre optique O et son plan image P_i (voir figure 2.1). Sans la moindre perte de généralité, la distance focale est fixée égale à 1 de telle sorte que tout point m ayant pour coordonnées $\underline{x} = (x \ y \ z)^T$ se projette par projection perspective sur le plan image en M de coordonnées $\underline{X} = (X \ Y \ 1)^T$ avec :

$$\underline{X} = \frac{1}{z} \underline{x} \quad (2.1)$$

Figure 2.1 : *Modèle de la caméra*

Considérons à présent une primitive géométrique paramétrable \mathcal{P}_s de la scène. Celle-ci est décrite par une équation du type :

$$h(\underline{x}, \underline{p}) = 0, \quad \forall \underline{x} \in \mathcal{P}_s \quad (2.2)$$

où h définit la nature de la primitive et \underline{p} sa configuration.

Notre objectif est d'estimer la valeur des paramètres \underline{p} afin de reconstruire et localiser la primitive \mathcal{P}_s décrite par h (le type de la primitive à reconstruire est donc supposé connu a priori). Pour ce faire, la représentation de \mathcal{P}_s par l'ensemble des paramètres \underline{p} est choisie complète et unique (quitte à rajouter si nécessaire des contraintes liant certains paramètres \underline{p} entre eux). Ainsi, quelle que soit la configuration de la primitive, il n'existera toujours qu'un seul jeu unique de valeurs de \underline{p} correspondant à cette configuration. Nous noterons par ailleurs n le nombre de paramètres indépendants de \underline{p} .

En divisant l'équation (2.2) par z (> 0) et en utilisant (2.1), l'équation (2.2) devient :

$$h'(\underline{X}, 1/z, \underline{p}) = 0 \quad (2.3)$$

Le théorème des fonctions implicites assure, sous la condition $\frac{\partial h'}{\partial (1/z)} \neq 0$, satisfaite pour tous les cas non dégénérés (un cas dégénéré se produit par exemple quand une droite se projette dans l'image en un point ou un cercle en un segment), qu'il existe une fonction unique μ autour d'une solution de (2.3) telle que :

$$1/z = \mu(\underline{X}, \underline{p}_0) \quad (2.4)$$

où les paramètres \underline{p}_0 , fonction des paramètres \underline{p} , sont choisis pour les mêmes raisons que précédemment de telle sorte que la dimension n_0 de \underline{p}_0 soit minimale ($n_0 < n$).

Soit \mathcal{P}_i la projection de \mathcal{P}_s sur le plan image. En utilisant (2.4) dans (2.3), on peut caractériser \mathcal{P}_i par une équation de la forme :

$$h'(\underline{X}, \underline{p}) = 0, \quad \forall \underline{X} \in \mathcal{P}_i \quad (2.5)$$

qui, après reparamétrisation, peut se réécrire :

$$g(\underline{X}, \underline{P}) = 0, \quad \forall \underline{X} \in \mathcal{P}_i \quad (2.6)$$

où \underline{P} , fonction des paramètres \underline{p} , est choisi de telle sorte que sa dimension m soit minimale ($m < n$) [CHA90].

- **Remarque :** Pour une primitive plane, la fonction μ représente le plan contenant la primitive. Pour une primitive tridimensionnelle (sphère, cylindre, tore,...), la fonction $g(\underline{X}, \underline{P})$ représente l'équation des limbes et définit le contour dans l'image de la projection de la primitive. La relation entre ces points contour et les points 3D correspondants s'exprime par la fonction $\mu(\underline{X}, \underline{p}_0)$ qui représente alors la surface des limbes.

Soit $T = (V(O), \Omega)$ le torseur cinématique de la caméra où $V(O) = (V_x V_y V_z)$ représente sa vitesse de translation et $\Omega = (\Omega_x \Omega_y \Omega_z)^T$ sa vitesse de rotation.

La variation de \underline{P} qui permet de lier le mouvement apparent de la primitive dans l'image au mouvement de la caméra peut alors s'exprimer par l'équation fondamentale [CHA90] [ESP92] :

$$\dot{\underline{P}} = L_{\underline{P}}^T(\underline{P}, \underline{p}_0) T \quad (2.7)$$

où $L_{\underline{P}}^T$, appelée matrice d'interaction associée à \underline{P} , caractérise entièrement les interactions entre la caméra et la primitive considérée. Dans [CHA90] et [ESP92] est décrite une méthode de calcul de la matrice d'interaction associée aux primitives géométriques paramétrables. Les résultats explicites y sont donnés pour les primitives les plus usuelles : point, segment, droite, cercle, sphère et cylindre.

2.3 Reconstruction 3D de primitives géométriques

La méthode de reconstruction 3D que nous allons présenter est directement liée au formalisme qui vient d'être rappelé, à savoir la matrice d'interaction associée à une

primitive.

Soit $\mathcal{H}(\underline{P}, \dot{\underline{P}}, \underline{p}_0, T)$ la fonctionnelle suivante définie à partir de (2.7) :

$$\mathcal{H}(\underline{P}, \dot{\underline{P}}, \underline{p}_0, T) = \dot{\underline{P}} - L_{\underline{P}}^T(\underline{P}, \underline{p}_0) T = 0 \quad (2.8)$$

Sous la condition $\frac{\partial \mathcal{H}}{\partial \underline{p}_0}$ (de dimension $m * n_0$) de rang plein n_0 , le théorème des fonctions implicites nous permet d'exprimer \underline{p}_0 en fonction des autres variables de la fonctionnelle (2.8) :

$$\underline{p}_0 = \underline{p}_0(T, \underline{P}, \dot{\underline{P}}) \quad (2.9)$$

Généralement, comme nous le verrons dans les exemples décrits par la suite, les paramètres \underline{p}_0 s'obtiennent simplement par résolution d'un système linéaire.

Il est par ailleurs intéressant de noter que la condition d'utilisation du théorème des fonctions implicites permet de détecter, lorsqu'elle n'est pas vérifiée, les configurations de la primitive et les mouvements de la caméra qui ne permettent pas de remonter à une information tridimensionnelle sur cette primitive.

Connaissant donc la structure du torseur d'interaction associé à la primitive à reconstruire et ayant mesuré le long d'une séquence de deux images la vitesse de la caméra T , l'ensemble des paramètres 2D \underline{P} associés à la primitive \mathcal{P}_i et leur variation $\dot{\underline{P}}$, on peut déterminer les paramètres \underline{p}_0 qui représentent la fonction μ donnée par (2.4). La profondeur de tous les points de la primitive dont la projection dans l'image vérifie $g(\underline{X}, \underline{P}) = 0$ est alors connue. On peut ensuite calculer les paramètres \underline{p} qui définissent complètement la configuration de la primitive :

- **Cas des primitives planes** : La fonction μ représente dans ce cas le plan contenant la primitive. Les paramètres \underline{p} sont obtenus en effectuant l'intersection de ce plan avec le cône généralisé défini par son sommet au centre optique de la caméra O et la fonction $g(\underline{X}, \underline{P})$.
- **Cas des primitives tri-dimensionnelles** : L'intersection du cône défini par O et g avec la surface des limbes définie par μ permet la reconstruction 3D des limbes (voir figure 2.2). Des propriétés spécifiques à celles-ci telles que la tangence (voir par exemple le cas du cylindre présenté au paragraphe 2.3.5) permettent ensuite de déterminer l'ensemble des paramètres \underline{p} .

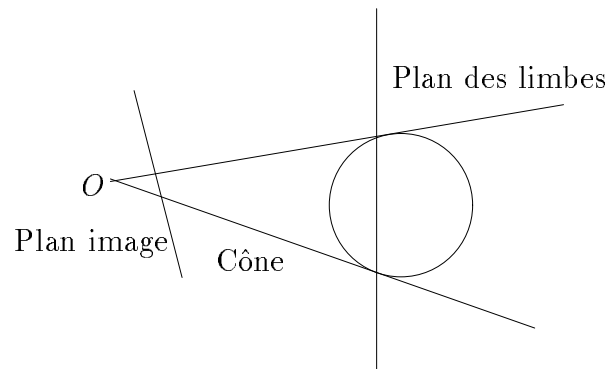


Figure 2.2 : *Vue de dessus d'un cylindre : approche continue*

- **Remarque :** L'approche discrète, similaire à celle de la reconstruction 3D par stéréovision, est basée sur la triangulation. Elle consiste à calculer l'intersection de deux cônes généralisés (un pour chaque position de la caméra), soit l'intersection de deux volumes (voir figure 2.3). L'approche continue consiste, elle, à effectuer l'intersection d'un volume et d'une surface (un plan ou la surface des limbes selon les cas) et s'avère donc de complexité moindre.

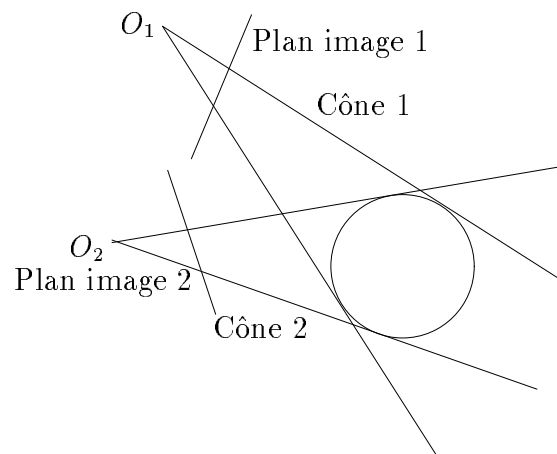


Figure 2.3 : *Vue de dessus d'un cylindre : approche discrète*

Nous allons maintenant nous attacher à reconstruire par la méthode que nous venons de décrire les primitives géométriques qui nous ont paru les plus signifi-

tives à savoir le point, la droite, le cercle, la sphère et le cylindre. Signalons que la reconstruction des primitives de type point et segment par la même approche a déjà fait l'objet de quelques travaux [ESP87] [XIE89].

2.3.1 Les points

Considérons un point de coordonnées $(x_0, y_0, z_0)^T$ dans le repère de la caméra. Dans ce cas très simple, on a :

$$h(\underline{x}, \underline{p}) : \begin{cases} x - x_0 = 0 \\ y - y_0 = 0 \\ z - z_0 = 0 \end{cases}, \quad g(\underline{X}, \underline{P}) : \begin{cases} X - X_0 = 0 \\ Y - Y_0 = 0 \end{cases}, \quad \mu(\underline{X}, \underline{p}_0) : 1/z = 1/z_0 \quad (2.10)$$

avec $\underline{p} = (x_0, y_0, z_0)$, $\underline{P} = (X_0, Y_0) = (x_0/z_0, y_0/z_0)$ et $\underline{p}_0 = 1/z_0$.

Les variations des coordonnées d'un point dans l'image en fonction du torseur cinématique de la caméra (appelées également équations d'"optic flow") sont écrites ici sous la forme :

$$\begin{pmatrix} \dot{X}_0 \\ \dot{Y}_0 \end{pmatrix} = L_{\underline{P}}^T T \quad (2.11)$$

avec [WAX87], [CHA90] :

$$L_{\underline{P}}^T = \begin{pmatrix} -1/z_0 & 0 & X_0/z_0 & X_0 Y_0 & -(1 + X_0^2) & Y_0 \\ 0 & -1/z_0 & Y_0/z_0 & 1 + Y_0^2 & -X_0 Y_0 & -X_0 \end{pmatrix} \quad (2.12)$$

L'équation (2.11) permet de calculer la profondeur z_0 du point. En effet, on peut la réécrire sous la forme :

$$A \ 1/z_0 = B \quad (2.13)$$

avec :

$$A = \begin{pmatrix} V_x - X_0 V_z \\ V_y - Y_0 V_z \end{pmatrix}, \quad B = \begin{pmatrix} \alpha_x \\ \alpha_y \end{pmatrix} = \begin{pmatrix} X_0 Y_0 \Omega_x - (1 + X_0^2) \Omega_y + Y_0 \Omega_z - \dot{X}_0 \\ (1 + Y_0^2) \Omega_x - X_0 Y_0 \Omega_y - X_0 \Omega_z - \dot{Y}_0 \end{pmatrix}$$

La solution aux moindres carrés est alors donnée par $1/z_0 = (A^T A)^{-1} A^T B$ sous réserve que $A^T A$ soit inversible et on obtient :

$$1/z_0 = \frac{\alpha_x (X_0 V_z - V_x) + \alpha_y (Y_0 V_z - V_y)}{(X_0 V_z - V_x)^2 + (Y_0 V_z - V_y)^2} \quad (2.14)$$

- **Remarque :** La condition $\frac{\partial \mathcal{H}}{\partial \underline{p}_0}$ de rang plein est ici équivalente à $A^T A$ inversible. On en déduit les mouvements bien connus de la caméra qui ne permettent pas d'estimer la profondeur z_0 d'un point par vision dynamique et qui consistent à avancer dans la direction du point : $V_x = X_0 V_z$ et $V_y = Y_0 V_z$.

2.3.2 Les droites

Nous représentons une droite par un vecteur $\underline{p} = (a_1, b_1, c_1, a_2, b_2, c_2, d_2)$ tel que les points de coordonnées (x, y, z) appartenant à la droite vérifient :

$$h(\underline{x}, \underline{p}) = \begin{cases} a_1 x + b_1 y + c_1 z = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \end{cases} \quad (2.15)$$

Une droite est ainsi représentée comme l'intersection de deux plans \mathcal{P}_1 et \mathcal{P}_2 , l'un deux passant par le centre de projection. Cette représentation n'est pas minimale mais elle a l'avantage d'être complète. Elle permet en effet de décrire n'importe quelle droite de l'espace 3D.

Pour que la représentation soit unique et non ambiguë, le choix de trois contraintes s'impose (quatre paramètres sont suffisants pour représenter une droite). On choisit bien évidemment les contraintes d'orthonormalité suivantes :

$$\begin{cases} a_1^2 + b_1^2 + c_1^2 & = 1 \\ a_2^2 + b_2^2 + c_2^2 & = 1 \\ a_1 a_2 + b_1 b_2 + c_1 c_2 & = 0 \end{cases} \quad (2.16)$$

Notons qu'on ne s'intéresse pas au cas dégénéré où la droite passe par le centre de projection ($d_2 = 0$) et qui donne un point pour image de la droite.

La fonction μ définie par (2.4) s'obtient facilement en utilisant l'équation de projection perspective (2.1) dans (2.15) :

$$\mu(\underline{X}, \underline{p}_0) : 1/z = aX + bY + c \quad (2.17)$$

avec $\underline{p}_0 = (a, b, c) = (-a_2/d_2, -b_2/d_2, -c_2/d_2)$.

Par ailleurs, la projection dans l'image de la droite s'exprime par l'équation :

$$a_1 X + b_1 Y + c_1 = 0 \quad (2.18)$$

qui peut se réécrire en utilisant la représentation minimale $\underline{P} = (\rho, \theta)$ (voir figure 2.4) :

$$g(\underline{X}, \underline{P}) = X \cos \theta + Y \sin \theta - \rho = 0 \quad (2.19)$$

avec :

$$\begin{cases} \cos \theta &= a_1 / \sqrt{a_1^2 + b_1^2} \\ \sin \theta &= b_1 / \sqrt{a_1^2 + b_1^2} \\ \rho &= -c_1 / \sqrt{a_1^2 + b_1^2} \end{cases} \quad (2.20)$$

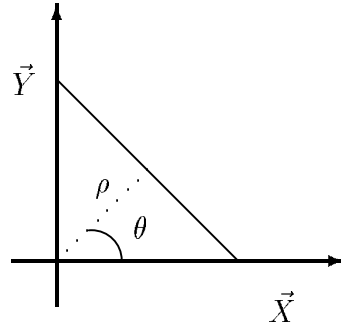


Figure 2.4 : Représentation (ρ, θ) des droites 2D

- **Remarque** : Les ambiguïtés liées à cette représentation (les jeux de paramètres $(\rho, \theta + 2k\pi)$ et $(-\rho, \theta + (2k + 1)\pi)$, $\forall k$, représentent tous la même droite) sont facilement levées en choisissant pour θ l'orientation modulée à 2π entre l'axe \vec{X} du plan image et la normale de la droite, orientée par convention dans le sens de la diminution du gradient de l'intensité lumineuse.

Les paramètres a_1, b_1 et c_1 du plan \mathcal{P}_1 passant par le centre de projection se calculent immédiatement à partir des valeurs de ρ et θ . On obtient, puisque $a_1^2 + b_1^2 + c_1^2$ a été fixé égal à 1 :

$$\begin{cases} a_1 &= \cos \theta / \sqrt{1 + \rho^2} \\ b_1 &= \sin \theta / \sqrt{1 + \rho^2} \\ c_1 &= -\rho / \sqrt{1 + \rho^2} \end{cases} \quad (2.21)$$

Pour calculer les paramètres du second plan caractérisant la droite à reconstruire, nous devons, comme évoqué précédemment, utiliser la matrice d'interaction $L_{\underline{P}}^T$ associé à $\underline{P} = (\rho, \theta)$. Celle-ci s'écrit [CHA90] :

$$L_{\underline{P}}^T = \begin{pmatrix} \lambda_\rho \cos \theta & \lambda_\rho \sin \theta & -\lambda_\rho \rho & (1 + \rho^2) \sin \theta & -(1 + \rho^2) \cos \theta & 0 \\ \lambda_\theta \cos \theta & \lambda_\theta \sin \theta & -\lambda_\theta \rho & -\rho \cos \theta & -\rho \sin \theta & -1 \end{pmatrix} \quad (2.22)$$

avec :

$$\begin{cases} \lambda_\rho &= -(a\rho \cos \theta + b\rho \sin \theta + c) \\ \lambda_\theta &= -a \sin \theta + b \cos \theta \end{cases} \quad (2.23)$$

Les paramètres $\underline{p}_0 = (a, b, c)$ sont obtenus en résolvant le système linéaire suivant construit à partir de (2.23) et de la contrainte d'orthogonalité entre les plans \mathcal{P}_1 et \mathcal{P}_2 :

$$\begin{cases} -a\rho \cos \theta - b\rho \sin \theta - c &= \lambda_\rho \\ -a \sin \theta + b \cos \theta &= \lambda_\theta \\ a \cos \theta + b \sin \theta - c\rho &= 0 \end{cases} \quad (2.24)$$

On en déduit :

$$\begin{cases} a &= -((1 + \rho^2)\lambda_\theta \sin \theta + \rho\lambda_\rho \cos \theta)/(1 + \rho^2) \\ b &= ((1 + \rho^2)\lambda_\theta \cos \theta - \rho\lambda_\rho \sin \theta)/(1 + \rho^2) \\ c &= -\lambda_\rho/(1 + \rho^2) \end{cases} \quad (2.25)$$

où λ_ρ et λ_θ sont obtenus à partir de l'équation (2.7) et de la matrice d'interaction associée aux droites (2.22) :

$$\begin{cases} \lambda_\rho &= \frac{\dot{\rho} + (1 + \rho^2)(\Omega_y \cos \theta - \Omega_x \sin \theta)}{V_x \cos \theta + V_y \sin \theta - \rho V_z} \\ \lambda_\theta &= \frac{\dot{\theta} + \rho(\Omega_x \cos \theta + \Omega_y \sin \theta) + \Omega_z}{V_x \cos \theta + V_y \sin \theta - \rho V_z} \end{cases} \quad (2.26)$$

Enfin, on obtient $d_2 = 1/\sqrt{a^2 + b^2 + c^2}$, $a_2 = -ad_2$, $b_2 = -bd_2$ et $c_2 = -cd_2$. L'ensemble des coefficients représentant la droite est ainsi complètement caractérisé.

- **Remarque** : On peut facilement montrer en étudiant le rang de la matrice $\frac{\partial \mathcal{H}}{\partial \underline{p}_0}$ définie à partir de (2.8) que les mouvements de la caméra ne permettant pas de reconstruire une droite sont tels que $V_x \cos \theta + V_y \sin \theta - \rho V_z = 0$, soit des mouvements appartenant au plan \mathcal{P}_1 passant par le centre optique et la droite.

2.3.3 Les cercles

Un cercle peut être représenté comme l'intersection d'une sphère et d'un plan coupant la sphère en son centre. On a alors :

$$h(\underline{x}, \underline{p}) = \begin{cases} h_1 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \\ h_2 = \alpha(x - x_0) + \beta(y - y_0) + \gamma(z - z_0) = 0 \end{cases} \quad (2.27)$$

Un cercle est ainsi caractérisé par un vecteur $\underline{p} = (x_0, y_0, z_0, r, \alpha, \beta, \gamma)$ de dimension 7 où r est le rayon du cercle, x_0, y_0 et z_0 sont les coordonnées de son centre et α, β et γ sont les composantes de la normale au cercle. Cette représentation est complète mais non minimale (six paramètres sont suffisants pour représenter un cercle). On rajoute donc la contrainte suivante :

$$\alpha^2 + \beta^2 + \gamma^2 = 1 \quad (2.28)$$

La fonction μ définie par (2.4) est obtenue en substituant (2.1) dans h_2 :

$$\mu(\underline{X}, \underline{p}_0) : 1/z = aX + bY + c \quad (2.29)$$

avec :

$$\begin{cases} a = \alpha/(\alpha x_0 + \beta y_0 + \gamma z_0) \\ b = \beta/(\alpha x_0 + \beta y_0 + \gamma z_0) \\ c = \gamma/(\alpha x_0 + \beta y_0 + \gamma z_0) \end{cases} \quad (2.30)$$

L'image d'un cercle est généralement une ellipse (parfois un cercle) et est définie par une équation du type [CHA90] :

$$g(\underline{X}, \underline{P}) = K_0 X^2 + K_1 Y^2 + 2K_2 XY + 2K_3 X + 2K_4 Y + K_5 = 0 \quad (2.31)$$

avec :

$$\begin{cases} K_0 = a^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2ax_0 \\ K_1 = b^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2by_0 \\ K_2 = ab(x_0^2 + y_0^2 + z_0^2 - r^2) - bx_0 - ay_0 \\ K_3 = ac(x_0^2 + y_0^2 + z_0^2 - r^2) - cx_0 - az_0 \\ K_4 = bc(x_0^2 + y_0^2 + z_0^2 - r^2) - cy_0 - bz_0 \\ K_5 = c^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2cz_0 \end{cases} \quad (2.32)$$

- **Remarque :** Pour le cercle, les cas dégénérés sont tels que le plan contenant le cercle passe par le centre optique ($\alpha x_0 + \beta y_0 + \gamma z_0 = 0$) : l'image d'un cercle est alors un segment.

Un paramétrage minimal d'une ellipse \mathcal{E} est obtenu à partir des moments $m_{ij} = \sum_{X \in \mathcal{E}} \sum_{Y \in \mathcal{E}} X^i Y^j$ d'ordre inférieur à 3 ($i + j \leq 2$). On choisit les moments centrés, facilement mesurables dans l'image, $\underline{P} = (X_c, Y_c, \mu_{20}, \mu_{11}, \mu_{02})$ et on peut établir les relations suivantes :

$$\begin{cases} X_c = m_{10}/m_{00} & = (K_1 K_3 - K_2 K_4)/\Delta \\ Y_c = m_{01}/m_{00} & = (K_0 K_4 - K_2 K_3)/\Delta \\ \mu_{20} = 4(m_{20} - m_{00} X_c^2)/m_{00} & = -K_1 K/\Delta \\ \mu_{11} = 4(m_{11} - m_{00} X_c Y_c)/m_{00} & = K_2 K/\Delta \\ \mu_{02} = 4(m_{02} - m_{00} Y_c^2)/m_{00} & = -K_0 K/\Delta \end{cases} \quad (2.33)$$

avec $\Delta = K_2^2 - K_0K_1$ et $K = K_0X_c^2 + 2K_2X_cY_c + K_1Y_c^2 - K_5$.

La matrice d'interaction associée à cette représentation s'écrit [CHA90] :

$$L_{\underline{P}}^T = \begin{pmatrix} -1/z_c & 0 & X_c/z_c + a\mu_{20} + b\mu_{11} \\ X_cY_c + \mu_{11} & -1 - X_c^2 - \mu_{20} & Y_c \\ 0 & -1/z_c & Y_c/z_c + a\mu_{11} + b\mu_{02} \\ 1 + Y_c^2 + \mu_{02} & -X_cY_c - \mu_{11} & -X_c \\ -2(a\mu_{20} + b\mu_{11}) & 0 & 2[(1/z_c + aX_c)\mu_{20} + bX_c\mu_{11}] \\ 2(Y_c\mu_{20} + X_c\mu_{11}) & -4\mu_{20}X_c & 2\mu_{11} \\ -a\mu_{11} - b\mu_{02} & -a\mu_{20} - b\mu_{11} & aY_c\mu_{20} + (3/z_c - c)\mu_{11} + bX_c\mu_{02} \\ 3Y_c\mu_{11} + X_c\mu_{02} & -Y_c\mu_{20} - 3X_c\mu_{11} & \mu_{02} - \mu_{20} \\ 0 & -2(a\mu_{11} + b\mu_{02}) & 2[(1/z_c + bY_c)\mu_{02} + aY_c\mu_{11}] \\ 4Y_c\mu_{02} & -2(Y_c\mu_{11} + X_c\mu_{02}) & -2\mu_{11} \end{pmatrix} \quad (2.34)$$

où $z_c = 1/(aX_c + bY_c + c)$.

Les paramètres $\underline{p}_0 = (a, b, c)$ sont déterminés directement par la résolution aux moindres carrés du système linéaire défini à partir de (2.34). On en déduit en utilisant (2.30) et (2.28) les paramètres α, β et γ , composantes de la normale au plan du cercle. On obtient :

$$\begin{cases} \alpha = \pm a/\sqrt{a^2 + b^2 + c^2} \\ \beta = \pm b/\sqrt{a^2 + b^2 + c^2} \\ \gamma = \pm c/\sqrt{a^2 + b^2 + c^2} \end{cases} \quad (2.35)$$

- **Remarque :** L'ambiguïté logiquement observée sur le sens de la normale au plan du cercle peut facilement être levée en se fixant une contrainte choisie par convention (par exemple $\alpha x_0 + \beta y_0 + \gamma z_0 > 0$).

Considérons à présent le système linéaire suivant construit à partir de (2.30) et (2.32) :

$$\begin{cases} ax_0 + by_0 + cz_0 & = 1 \\ (acK_0 - a^2K_3)A - (cK_0 - 2aK_3)x_0 - aK_0z_0 & = K_3 \\ (bcK_0 - a^2K_4)A + 2aK_4x_0 - cK_0y_0 - bK_0z_0 & = K_4 \\ (c^2K_0 - a^2K_5)A + 2aK_5x_0 - 2cK_0z_0 & = K_5 - K_0 \\ (b^2K_0 - a^2K_1)A + 2aK_1x_0 - 2bK_0y_0 & = K_1 - K_0 \\ (abK_0 - a^2K_2)A - (bK_0 - 2aK_2)x_0 - aK_0y_0 & = K_2 \end{cases} \quad (2.36)$$

avec $A = x_0^2 + y_0^2 + z_0^2 - r^2$. Comme la représentation d'une ellipse par les six paramètres K_i n'est pas minimale, leur valeur est impossible à déterminer de manière unique à partir des valeurs mesurées dans l'image des moments centrés. Par contre, si on pose $A_i = K_i/K_0$, ($i = 1, 5$) sachant que K_0 est toujours différent de 0 pour les cas non dégénérés, ce système se réécrit :

$$\begin{cases} ax_0 + by_0 + cz_0 & = 1 \\ (ac - a^2 A_3)A - (c - 2a A_3)x_0 - az_0 & = A_3 \\ (bc - a^2 A_4)A + 2a A_4 x_0 - cy_0 - bz_0 & = A_4 \\ (c^2 - a^2 A_5)A + 2a A_5 x_0 - 2cz_0 & = A_5 - 1 \\ (b^2 - a^2 A_1)A + 2a A_1 x_0 - 2by_0 & = A_1 - 1 \\ (ab - a^2 A_2)A - (b - 2a A_2)x_0 - ay_0 & = A_2 \end{cases} \quad (2.37)$$

où les paramètres A_i sont calculés à partir du changement de variable suivant :

$$\begin{cases} A_1 & = \mu_{20}/\mu_{02} \\ A_2 & = -\mu_{11}/\mu_{02} \\ A_3 & = (\mu_{11}Y_c - \mu_{02}X_c)/\mu_{02} \\ A_4 & = (\mu_{11}X_c - \mu_{20}Y_c)/\mu_{02} \\ A_5 & = (\mu_{02}X_c^2 + \mu_{20}Y_c^2 - 2\mu_{11}X_cY_c + \mu_{11}^2 - \mu_{20}\mu_{02})/\mu_{02} \end{cases} \quad (2.38)$$

On peut ainsi calculer A et les expressions des coordonnées x_0, y_0 et z_0 du centre du cercle par résolution aux moindres carrés du système linéaire (2.37). On en déduit ensuite la valeur du rayon du cercle :

$$r = \sqrt{x_0^2 + y_0^2 + z_0^2 - A} \quad (2.39)$$

Notre méthode se démarque ainsi des approches discrètes classiques par sa simplicité. Les paramètres du cercle sont déterminés en résolvant deux systèmes linéaires alors que la méthode discrète présentée dans [SAF92] repose sur une résolution non linéaire complexe (voir chapitre 1).

2.3.4 Les sphères

L'équation paramétrique d'une sphère de centre m_0 de coordonnées (x_0, y_0, z_0) et de rayon r s'écrit :

$$h(\underline{x}, \underline{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \quad (2.40)$$

Comme pour le cas du cercle, l'image d'une sphère est caractérisée par l'équation d'une ellipse [CHA90] :

$$K_0X^2 + K_1Y^2 + 2K_2XY + 2K_3X + 2K_4Y + K_5 = 0 \quad (2.41)$$

$$\text{avec } \begin{cases} K_0 = r^2 - y_0^2 - z_0^2 \\ K_1 = r^2 - x_0^2 - z_0^2 \\ K_2 = x_0y_0 \\ K_3 = x_0z_0 \\ K_4 = y_0z_0 \\ K_5 = r^2 - x_0^2 - y_0^2 \end{cases} \quad (2.42)$$

La fonction μ représentant la surface des limbes peut être déterminée par une démarche similaire à celle décrite plus loin dans le cas du cylindre (voir paragraphe 2.3.5). On obtient [CHA90] :

$$\mu(\underline{X}, \underline{p}_0) : 1/z = aX + bY + c \quad (2.43)$$

$$\text{avec } \begin{cases} a = x_0/(x_0^2 + y_0^2 + z_0^2 - r^2) \\ b = y_0/(x_0^2 + y_0^2 + z_0^2 - r^2) \\ c = z_0/(x_0^2 + y_0^2 + z_0^2 - r^2) \end{cases} \quad (2.44)$$

où $\underline{p}_0 = (a, b, c)$ représente ici encore la normale au plan des limbes de la sphère.

Comme pour le cas du cercle, on choisit une représentation de l'ellipse par les moments et les paramètres \underline{p}_0 sont obtenus par la résolution aux moindres carrés du système linéaire défini à partir de (2.34).

Les relations entre les paramètres $\underline{p} = (x_0, y_0, z_0, r)$ et $\underline{P} = (X_c, Y_c, \mu_{11}, \mu_{20}, \mu_{02})$ vont nous permettre, connaissant \underline{p}_0 , de déterminer \underline{p} . Ces relations sont données par :

$$\begin{cases} X_c = x_0z_0/(z_0^2 - r^2) \\ Y_c = y_0z_0/(z_0^2 - r^2) \\ \mu_{11} = x_0y_0r^2/(z_0^2 - r^2)^2 \\ \mu_{20} = r^2[X_c^2/z_0^2 + 1/(z_0^2 - r^2)] \\ \mu_{02} = r^2[Y_c^2/z_0^2 + 1/(z_0^2 - r^2)] \end{cases} \quad (2.45)$$

En additionnant deux à deux les deux dernières équations de (2.45), on obtient un polynôme du quatrième degré en z_0 :

$$z_0^4(\mu_{20} + \mu_{02}) + z_0^2r^2(\mu_{20} + \mu_{02} + X_c^2 + Y_c^2 + 2) + r^4(X_c^2 + Y_c^2) = 0 \quad (2.46)$$

dont le discriminant Δ peut s'écrire, après quelques développements, sous la forme :

$$\Delta = r^4 \Delta' \text{ avec } \Delta' = (\mu_{20} + \mu_{02} - X_c^2 - Y_c^2)^2 + 4(\mu_{20} + \mu_{02} + X_c^2 + Y_c^2) + 4 \quad (2.47)$$

Le discriminant Δ est strictement positif, (2.46) admet donc toujours deux solutions réelles qui sont :

$$z_0^2 = \lambda^2 r^2 \quad (2.48)$$

avec :

$$\lambda = \sqrt{\frac{\mu_{20} + \mu_{02} + X_c^2 + Y_c^2 + 2 \pm \sqrt{\Delta'}}{2(\mu_{20} + \mu_{02})}}$$

On peut vérifier que l'une des deux valeurs possibles de λ est inférieure à 1 quelle que soit la configuration de la sphère, l'autre étant par contre supérieure à 1. Nous rejetons bien évidemment la première solution puisque physiquement irréalisable (le centre de la sphère doit être devant le plan image $\Rightarrow \lambda > 1$).

On peut à présent calculer les paramètres $\underline{p} = (x_0, y_0, z_0, r)$ en utilisant les équations (2.44). On obtient :

$$\begin{cases} x_0 = az_0/c \\ y_0 = bz_0/c \\ z_0 = [(a^2 + b^2 + c^2)z_0^2 - c^2r^2]/c \end{cases} \quad (2.49)$$

D'où, en utilisant (2.48) dans cette dernière équation :

$$\begin{cases} r = c\lambda/[\lambda^2(a^2 + b^2 + c^2) - c^2] \\ z_0 = \lambda r \\ x_0 = a\lambda r/c \\ y_0 = b\lambda r/c \end{cases} \quad (2.50)$$

2.3.5 Les cylindres

2.3.5.1 Modélisation

L'équation paramétrique d'un cylindre s'écrit [CHA90] :

$$h(\underline{x}, \underline{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - (ax + by + cz)^2 - r^2 = 0 \quad (2.51)$$

où r est le rayon du cylindre, a , b et c les composantes du vecteur directeur de l'axe du cylindre et x_0 , y_0 et z_0 les coordonnées d'un point m_0 de cet axe. Un cylindre est donc

représenté par le vecteur $\underline{p} = (x_0, y_0, z_0, r, a, b, c)$. C'est un paramétrage complet mais non minimal. En effet, cinq paramètres sont suffisants pour représenter un cylindre. On rajoute donc les deux contraintes suivantes :

$$a^2 + b^2 + c^2 = 1 \quad (2.52)$$

$$ax_0 + by_0 + cz_0 = 0 \quad (2.53)$$

qui fixe notamment le point m_0 comme étant le point de l'axe du cylindre le plus proche du centre optique (voir figure 2.5).

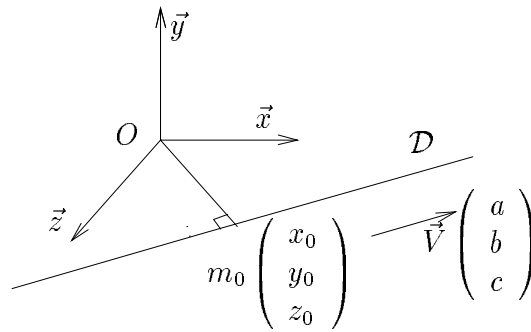


Figure 2.5 : Point le plus proche du centre optique

En appliquant l'équation de projection perspective (2.1) sur l'équation (2.51), on obtient un polynôme du second degré en $1/z$:

$$A 1/z^2 - 2 B 1/z + C = 0 \quad (2.54)$$

$$\text{avec } \begin{cases} A = x_0^2 + y_0^2 + z_0^2 - r^2 \\ B = x_0 X + y_0 Y + z_0 \\ C = X^2 + Y^2 + 1 - (aX + bY + c)^2 \end{cases}$$

Ce polynôme signifie que pour un point quelconque (X, Y) du plan image, il peut exister :

- soit aucun point d'intersection entre la ligne de vue associée à (X, Y) et le cylindre ;
- soit un point d'intersection ;
- soit deux points d'intersection.

Le cas qui nous intéresse est celui où il n'existe qu'un seul point d'intersection entre les lignes de vue des points du plan image et le cylindre. Ces points appartiennent aux contours (ou limbes) du cylindre. Dans le cas présent, cette propriété est équivalente au fait que le discriminant Δ de (2.54) soit nul.

Après quelques développements décrits dans [CHA90], on peut factoriser Δ sous la forme :

$$\Delta = [rB/\sqrt{A} - (\alpha X + \beta Y + \gamma)][rB/\sqrt{A} + (\alpha X + \beta Y + \gamma)] = 0 \quad (2.55)$$

$$\text{avec } \begin{cases} \alpha = cy_0 - bz_0 \\ \beta = az_0 - cx_0 \\ \gamma = bx_0 - ay_0 \end{cases}$$

qui donne l'équation de deux droites (une seule si le rayon du cylindre est nul). En utilisant la représentation (ρ, θ) des droites, on obtient :

$$\begin{aligned} D_1 & : X \cos \theta_1 + Y \sin \theta_1 - \rho_1 = 0 \\ D_2 & : X \cos \theta_2 + Y \sin \theta_2 - \rho_2 = 0 \end{aligned} \quad (2.56)$$

$$\text{avec } \begin{cases} \cos \theta_1 = \frac{rx_0/\sqrt{A}-\alpha}{\sqrt{(rx_0/\sqrt{A}-\alpha)^2+(ry_0/\sqrt{A}-\beta)^2}}, & \cos \theta_2 = \frac{rx_0/\sqrt{A}+\alpha}{\sqrt{(rx_0/\sqrt{A}+\alpha)^2+(ry_0/\sqrt{A}+\beta)^2}} \\ \sin \theta_1 = \frac{ry_0/\sqrt{A}-\beta}{\sqrt{(rx_0/\sqrt{A}-\alpha)^2+(ry_0/\sqrt{A}-\beta)^2}}, & \sin \theta_2 = \frac{ry_0/\sqrt{A}+\beta}{\sqrt{(rx_0/\sqrt{A}+\alpha)^2+(ry_0/\sqrt{A}+\beta)^2}} \\ \rho_1 = \frac{rz_0/\sqrt{A}-\gamma}{\sqrt{(rx_0/\sqrt{A}-\alpha)^2+(ry_0/\sqrt{A}-\beta)^2}}, & \rho_2 = \frac{rz_0/\sqrt{A}+\gamma}{\sqrt{(rx_0/\sqrt{A}+\alpha)^2+(ry_0/\sqrt{A}+\beta)^2}} \end{cases}$$

Par ailleurs, la fonction μ définie par (2.4) est facilement obtenue à partir de la racine double de (2.54) correspondant à $\Delta = 0$ et on obtient :

$$\mu(\underline{X}, \underline{p}_0) : 1/z = n_1 X + n_2 Y + n_3 \quad (2.57)$$

où $\underline{p}_0 = (n_1, n_2, n_3) = (x_0/A, y_0/A, z_0/A)$ sont les composantes de la normale au plan des limbes.

La matrice d'interaction $L_{\underline{P}}^T$ avec $\underline{P} = (\rho_1, \theta_1, \rho_2, \theta_2)$ s'écrit comme suit [CHA90] :

$$L_{\underline{P}}^T = \begin{pmatrix} \lambda_{\rho_1} \cos \theta_1 & \lambda_{\rho_1} \sin \theta_1 & -\lambda_{\rho_1} \rho_1 & (1 + \rho_1^2) \sin \theta_1 & -(1 + \rho_1^2) \cos \theta_1 & 0 \\ \lambda_{\theta_1} \cos \theta_1 & \lambda_{\theta_1} \sin \theta_1 & -\lambda_{\theta_1} \rho_1 & -\rho_1 \cos \theta_1 & -\rho_1 \sin \theta_1 & -1 \\ \lambda_{\rho_2} \cos \theta_2 & \lambda_{\rho_2} \sin \theta_2 & -\lambda_{\rho_2} \rho_2 & (1 + \rho_2^2) \sin \theta_2 & -(1 + \rho_2^2) \cos \theta_2 & 0 \\ \lambda_{\theta_2} \cos \theta_2 & \lambda_{\theta_2} \sin \theta_2 & -\lambda_{\theta_2} \rho_2 & -\rho_2 \cos \theta_2 & -\rho_2 \sin \theta_2 & -1 \end{pmatrix} \quad (2.58)$$

$$\text{avec } \begin{cases} \lambda_{\rho_1} &= -(n_1 \rho_1 \cos \theta_1 + n_2 \rho_1 \sin \theta_1 + n_3) \\ \lambda_{\theta_1} &= n_2 \cos \theta_1 - n_1 \sin \theta_1 \\ \lambda_{\rho_2} &= -(n_1 \rho_2 \cos \theta_2 + n_2 \rho_2 \sin \theta_2 + n_3) \\ \lambda_{\theta_2} &= n_2 \cos \theta_2 - n_1 \sin \theta_2 \end{cases} \quad (2.59)$$

2.3.5.2 Estimation des paramètres du cylindre à partir de deux limbes

Nous présentons ci-dessous une méthode d'estimation des paramètres du cylindre qui s'appuie sur l'observation de ses deux limbes sur deux images successives.

En résolvant le système linéaire (2.59), on peut facilement déterminer les paramètres $\underline{p}_0 = (n_1, n_2, n_3)$. On obtient :

$$\begin{cases} n_1 = \frac{(\rho_1 \sin \theta_1 - \rho_2 \sin \theta_2)(\lambda_{\theta_1} - \lambda_{\theta_2}) - (\cos \theta_2 - \cos \theta_1)(\lambda_{\rho_1} - \lambda_{\rho_2})}{(\rho_1 \cos \theta_1 - \rho_2 \cos \theta_2)(\cos \theta_2 - \cos \theta_1) - (\rho_1 \sin \theta_1 - \rho_2 \sin \theta_2)(\sin \theta_1 - \sin \theta_2)} \\ n_2 = \frac{(\sin \theta_1 - \sin \theta_2)(\lambda_{\rho_1} - \lambda_{\rho_2}) - (\rho_1 \cos \theta_1 - \rho_2 \cos \theta_2)(\lambda_{\theta_1} - \lambda_{\theta_2})}{(\rho_1 \cos \theta_1 - \rho_2 \cos \theta_2)(\cos \theta_2 - \cos \theta_1) - (\rho_1 \sin \theta_1 - \rho_2 \sin \theta_2)(\sin \theta_1 - \sin \theta_2)} \\ n_3 = \frac{\rho_1 \rho_2 (\lambda_{\theta_2} - \lambda_{\theta_1}) \sin(\theta_1 - \theta_2) + (1 - \cos(\theta_1 - \theta_2))(\rho_2 \lambda_{\rho_1} + \rho_1 \lambda_{\rho_2})}{(\rho_1 \cos \theta_1 - \rho_2 \cos \theta_2)(\cos \theta_2 - \cos \theta_1) - (\rho_1 \sin \theta_1 - \rho_2 \sin \theta_2)(\sin \theta_1 - \sin \theta_2)} \end{cases} \quad (2.60)$$

où λ_{ρ_i} et λ_{θ_i} ($i = 1, 2$) sont obtenus, comme pour le cas de la droite, à partir de l'équation (2.7) après y avoir substitué (2.58) :

$$\begin{cases} \lambda_{\rho_i} &= \frac{\dot{\rho}_i + (1 + \rho_i^2)(\Omega_y \cos \theta_i - \Omega_x \sin \theta_i)}{V_x \cos \theta_i + V_y \sin \theta_i - \rho_i V_z} \\ \lambda_{\theta_i} &= \frac{\dot{\theta}_i + \rho_i(\Omega_x \cos \theta_i + \Omega_y \sin \theta_i) + \Omega_z}{V_x \cos \theta_i + V_y \sin \theta_i - \rho_i V_z} \end{cases} \quad (2.61)$$

- **Remarque** : Dans le paragraphe suivant, nous décrivons une extension de cette méthode au cas plus restreint où un seul limbe du cylindre est observable. Dans ce cas, nous verrons qu'une image supplémentaire du cylindre s'impose pour compenser l'absence d'informations sur le second limbe. En effet, il est alors impossible de déterminer de manière unique les trois paramètres du plan des limbes puisque seules les deux premières équations du système (2.59) sont disponibles.

Connaissant les paramètres \underline{p}_0 , on peut alors déterminer l'ensemble des paramètres \underline{p} en se basant sur des contraintes géométriques liées au cylindre. Considérons en effet une vue de dessus du cylindre (voir figure 2.6) avec :

- P_i le plan image, P le plan des limbes, P_{t1} et P_{t2} les plans tangents au cylindre contenant respectivement les droites D_1 et D_2 , P_{n1} (respectivement P_{n2}) le plan orthogonal à P_{t1} (respectivement P_{t2}) passant par l'intersection de P_{t1} (respectivement P_{t2}) et P ;
- $\underline{V} = (a, b, c)$ le vecteur directeur de l'axe du cylindre, $\underline{N} = (n_1, n_2, n_3)$ la normale au plan des limbes P , $\underline{n}_{t1} = (\cos \theta_1, \sin \theta_1, -\rho_1)$ et $\underline{n}_{t2} = (\cos \theta_2, \sin \theta_2, -\rho_2)$ les normales à P_{t1} et P_{t2} , $\underline{n}_{n1} = (u_1, v_1, w_1)$ et $\underline{n}_{n2} = (u_2, v_2, w_2)$ les normales à P_{n1} et P_{n2} .

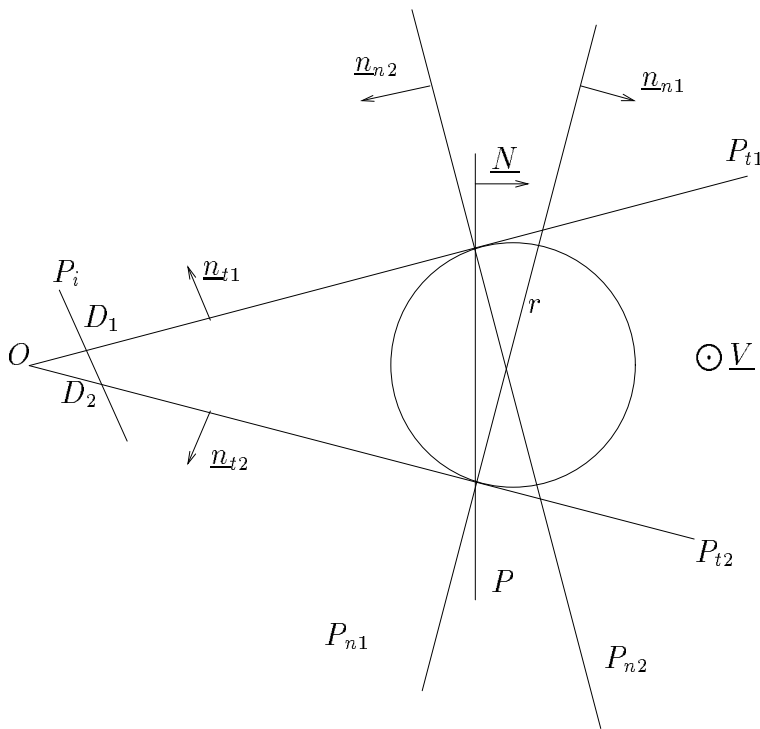


Figure 2.6 : Vue de dessus d'un cylindre

a) Estimation du vecteur directeur de l'axe du cylindre

Les coordonnées (a, b, c) de \underline{V} représentant la direction du cylindre se calculent en considérant le produit vectoriel suivant (voir figure 2.6) :

$$\underline{V} = \frac{\underline{N} \wedge (\underline{n}_{t1} - \underline{n}_{t2})}{\|\underline{N} \wedge (\underline{n}_{t1} - \underline{n}_{t2})\|} \quad (2.62)$$

On en déduit :

$$\begin{cases} a &= [n_2(\rho_2 - \rho_1) - n_3(\sin \theta_1 - \sin \theta_2)] / D \\ b &= [n_3(\cos \theta_1 - \cos \theta_2) - n_1(\rho_2 - \rho_1)] / D \\ c &= [n_1(\sin \theta_1 - \sin \theta_2) - n_2(\cos \theta_1 - \cos \theta_2)] / D \end{cases} \quad (2.63)$$

avec :

$$D = \|\underline{N} \wedge (\underline{n}_{t1} - \underline{n}_{t2})\|$$

b) Estimation de la position du cylindre et de son rayon

Les coordonnées (x_0, y_0, z_0) de m_0 , point de l'axe de révolution du cylindre le plus proche du centre optique, sont déterminées en utilisant la contrainte (2.53) et le fait que m_0 appartienne à l'intersection des deux plans P_{n_1} et P_{n_2} normaux au cylindre. On obtient une fois de plus un système linéaire :

$$\begin{cases} u_1 x_0 + v_1 y_0 + w_1 z_0 &= -l_1 \\ u_2 x_0 + v_2 y_0 + w_2 z_0 &= -l_2 \\ a x_0 + b y_0 + c z_0 &= 0 \end{cases} \quad (2.64)$$

où les paramètres (u_i, v_i, w_i, l_i) représentant le plan $P_{n_i}, i = 1, 2$ sont obtenus de la manière suivante :

- les coordonnées (u_i, v_i, w_i) de \underline{n}_{n_i} en considérant le produit vectoriel suivant (voir figure 2.6) :

$$\underline{n}_{n_i} = \underline{n}_{t_i} \wedge \underline{V} \Rightarrow \begin{cases} u_i &= b \rho_i + c \sin \theta_i \\ v_i &= -c \cos \theta_i - a \rho_i \\ w_i &= b \cos \theta_i - a \sin \theta_i \end{cases} \quad (2.65)$$

- l_i en calculant les coordonnées (x_i, y_i, z_i) du point particulier m_i appartenant à P_{n_i} tel que $\underline{m}_i \in \{P_{t_i} \cap P\}$ et vérifiant en outre, comme le point m_0 , la contrainte $a x_i + b y_i + c z_i = 0$. D'où :

$$\begin{cases} \cos \theta_i x_i + \sin \theta_i y_i - \rho_i z_i &= 0 \\ n_1 x_i + n_2 y_i + n_3 z_i &= 1 \\ a x_i + b y_i + c z_i &= 0 \end{cases} \quad (2.66)$$

dont la résolution donne :

$$\begin{cases} x_i = \frac{c \sin \theta_i + b \rho_i}{\cos \theta_i (bn_3 - cn_2) + \sin \theta_i (cn_1 - an_3) - \rho_i (an_2 - bn_1)} \\ y_i = \frac{-(c \cos \theta_i + a \rho_i)}{\cos \theta_i (bn_3 - cn_2) + \sin \theta_i (cn_1 - an_3) - \rho_i (an_2 - bn_1)} \\ z_i = \frac{b \cos \theta_i - a \sin \theta_i}{\cos \theta_i (bn_3 - cn_2) + \sin \theta_i (cn_1 - an_3) - \rho_i (an_2 - bn_1)} \end{cases} \quad (2.67)$$

On en déduit :

$$l_i = -(u_i x_i + v_i y_i + w_i z_i) \quad (2.68)$$

$$= -\frac{(c \sin \theta_i + b \rho_i)^2 + (c \cos \theta_i + a \rho_i)^2 + (b \cos \theta_i - a \sin \theta_i)^2}{\cos \theta_i (bn_3 - cn_2) + \sin \theta_i (cn_1 - an_3) - \rho_i (an_2 - bn_1)} \quad (2.69)$$

La valeur de x_0, y_0 et z_0 est alors donnée par :

$$\begin{cases} x_0 = -\frac{l_2(cv_1 - bw_1) + l_1(bw_2 - cv_2)}{u_1(bw_2 - cv_2) + v_1(cu_2 - aw_2) + w_1(av_2 - bu_2)} \\ y_0 = \frac{l_2(cu_1 - aw_1) + l_1(aw_2 - cu_2)}{u_1(bw_2 - cv_2) + v_1(cu_2 - aw_2) + w_1(av_2 - bu_2)} \\ z_0 = -\frac{l_2(bu_1 - av_1) + l_1(av_2 - bu_2)}{u_1(bw_2 - cv_2) + v_1(cu_2 - aw_2) + w_1(av_2 - bw_2)} \end{cases} \quad (2.70)$$

Finalement, la reconstruction du cylindre se complète par la détermination de son rayon r qui est donné par la distance entre les points m_0 et m_1 (ou m_2) :

$$r = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2} \quad (2.71)$$

2.3.5.3 Estimation des paramètres du cylindre à partir d'un seul limbe

Dans certaines situations, les deux limbes du cylindre ne sont pas tous les deux observables. Cela peut provenir par exemple de l'occlusion d'un des deux limbes ou éventuellement de contraintes de temps de calcul qui ne permettent le suivi que d'un

seul limbe dans les images successives¹. Dans ce cas, comme nous l'avons déjà vu, les paramètres du plan des limbes ne peuvent pas être déterminés de manière unique. L'utilisation du torseur d'interaction associé à l'unique limbe observé dans l'image nous permet cependant d'aboutir.

a) Estimation du vecteur directeur de l'axe du cylindre

On a à présent (voir figure 2.7) :

$$\underline{V} = \underline{N} \wedge \underline{n_{t1}} \quad (2.72)$$

soit :

$$\begin{cases} a &= -n_2\rho_1 - n_3 \sin \theta_1 \\ b &= n_3 \cos \theta_1 + n_1\rho_1 \\ c &= n_1 \sin \theta_1 - n_2 \cos \theta_1 \end{cases} \quad (2.73)$$

Les valeurs n_1, n_2 et n_3 représentant le plan des limbes sont dans le cas présent inconnues. On sait cependant d'après (2.59) que :

$$\begin{cases} n_1\rho_1 \cos \theta_1 + n_2\rho_1 \sin \theta_1 + n_3 &= -\lambda_{\rho_1} \\ n_2 \cos \theta_1 - n_1 \sin \theta_1 &= \lambda_{\theta_1} \end{cases} \quad (2.74)$$

où λ_{θ_1} et λ_{ρ_1} sont obtenues avec (2.61).

On en déduit, en combinant les deux premières équations de (2.73) :

$$\begin{cases} b \cos \theta_1 - a \sin \theta_1 &= -\lambda_{\rho_1} \\ c &= -\lambda_{\theta_1} \end{cases} \quad (2.75)$$

On sait par ailleurs que la normale $\underline{n_{t1}}$ du plan tangent passant par le limbe observé D_1 est par définition perpendiculaire au vecteur directeur de l'axe du cylindre. Ainsi, l'expression suivante doit être vérifiée :

$$\underline{n_{t1}} \cdot \underline{V} = 0 \quad (2.76)$$

On obtient finalement avec (2.75) et (2.76) un système linéaire à 3 équations et 3 inconnues dont la résolution donne :

$$\begin{cases} a &= -\rho_1 \lambda_{\theta_1} \cos \theta_1 + \lambda_{\rho_1} \sin \theta_1 \\ b &= -\rho_1 \lambda_{\theta_1} \sin \theta_1 + \lambda_{\rho_1} \cos \theta_1 \\ c &= -\lambda_{\theta_1} \end{cases} \quad (2.77)$$

Ces valeurs sont ensuite normalisées de telle sorte que $a^2 + b^2 + c^2 = 1$.

¹Nous reviendrons plus en détails sur les problèmes de traitement d'images dans le chapitre 4.

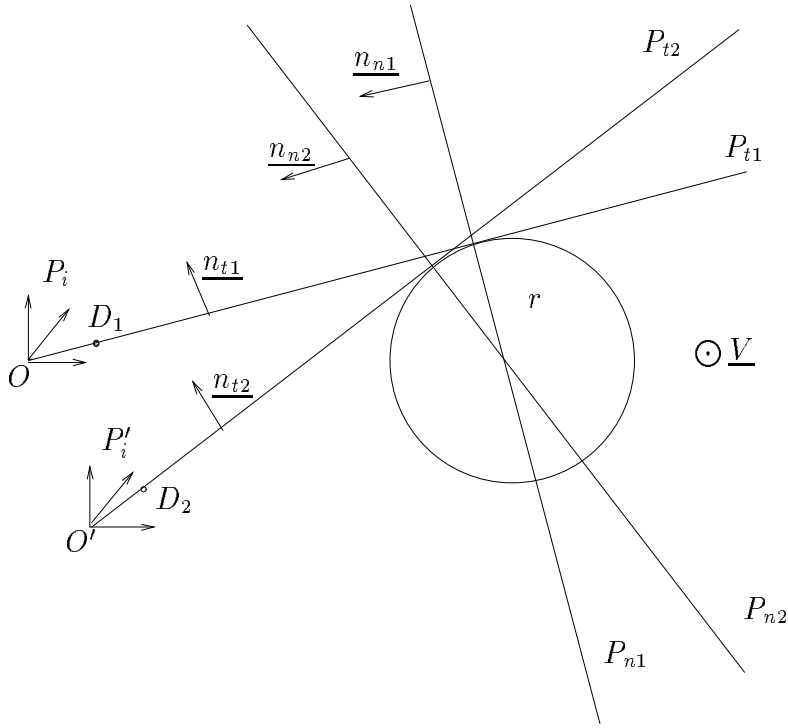


Figure 2.7 : Vue de dessus d'un limbe d'un cylindre avant et après déplacement de l'observateur

b) Estimation de la position du cylindre et de son rayon

Dans le cas précédent des deux limbes, la méthode proposée pour estimer les paramètres x_0, y_0, z_0 et r nécessite la connaissance des paramètres n_1, n_2 et n_3 . Ceux-ci sont en effet utilisés pour calculer les paramètres l_i des plans P_{ni} normaux au cylindre et passant par les deux limbes. Il se pose donc ici le même problème que celui rencontré pour l'estimation des paramètres du vecteur directeur de l'axe du cylindre.

Ici encore, toujours en s'appuyant sur les équations du torseur d'interaction lié au seul limbe considéré (voir (2.59)), il est possible de réécrire les paramètres $l_i (i = 1, 2)$ sous la forme suivante :

$$l_i = -\frac{(c \sin \theta_i + b \rho_i)^2 + (c \cos \theta_i + a \rho_i)^2 + (b \cos \theta_i - a \sin \theta_i)^2}{a(\lambda_{\rho_i} \sin \theta_i - \rho_i \lambda_{\theta_i} \cos \theta_i) - b(\lambda_{\rho_i} \cos \theta_i + \lambda_{\theta_i} \rho_i \sin \theta_i) - c \lambda_{\theta_i}} \quad (2.78)$$

L'utilisation d'un seul limbe ne nous permet cependant que de calculer l'équation d'un seul plan P_{n1} . Pour obtenir l'équation d'un second plan P_{n2} , on utilise une

troisième image : la première et la deuxième images fournissent l'équation du plan P_{n1} , la deuxième et la troisième images fournissent l'équation du plan P_{n2} . Il faut en outre transformer les paramètres du plan P_{n2} exprimés dans le repère caméra lié au deuxième point de vue dans le repère caméra lié au premier point de vue (voir figure 2.7). Cette transformation est bien évidemment déterminée directement à partir du torseur cinématique de la caméra (voir l'annexe A qui traite des transformations homogènes).

L'utilisation de P_{n1} avec P_{n2} , plus la prise en compte de la contrainte (2.53) permet, comme pour le cas de deux limbes, de calculer x_0, y_0, z_0 et r .

2.3.6 Résultats de simulation

La méthode de reconstruction 3D décrite dans le paragraphe 2.3 a été testée en simulation sur toutes les primitives qui ont été évoquées précédemment : point, droite, cercle, sphère et cylindre. Nous ne présentons ici que les résultats portant sur la reconstruction d'un point, d'une sphère et d'un cylindre.

2.3.6.1 Exemple 1 : le point

La figure 2.8 illustre dans sa partie gauche la position initiale de la caméra par rapport au point. A droite est représentée l'image correspondante.

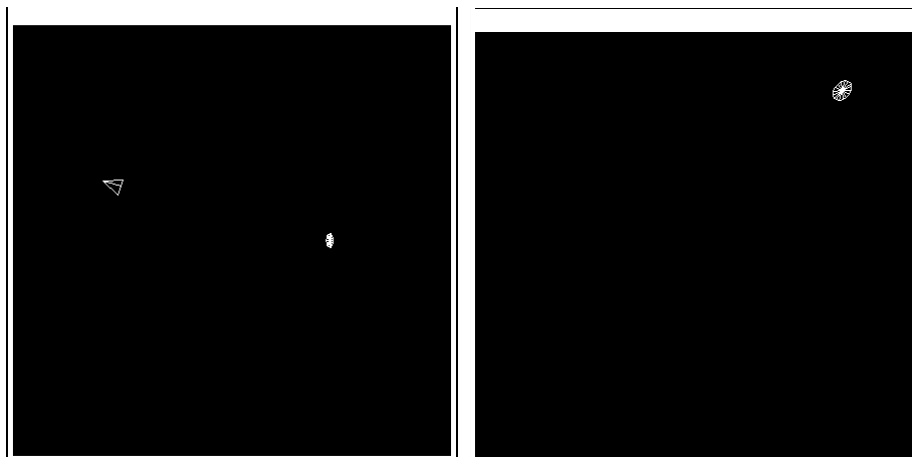


Figure 2.8 : *Position entre la caméra et le point*

A partir de cette position, on applique un mouvement quelconque à la caméra

(en l'occurrence $V_x = V_y = V_z = 0.5$ cm/s, $\Omega_x = \Omega_y = \Omega_z = 3$ dg/s) et, à chaque nouvelle acquisition d'image, on mesure les paramètres $\underline{P} = (X, Y)$ du point observé dans l'image, leur déplacement dans l'image $\Delta \underline{P}$ par rapport à leur valeur précédente ainsi que le déplacement de la caméra. On est alors en mesure d'estimer à chaque itération les coordonnées (x_0, y_0, z_0) du point 3D. Les résultats obtenus sont illustrés sur la figure 2.9 : à gauche, les différentes valeurs de (x_0, y_0, z_0) en cm calculées à chaque itération et exprimées dans un repère de référence fixe (les valeurs devraient donc être constantes puisque le point est fixe) et, à droite, l'erreur entre ces valeurs estimées et les valeurs réelles.

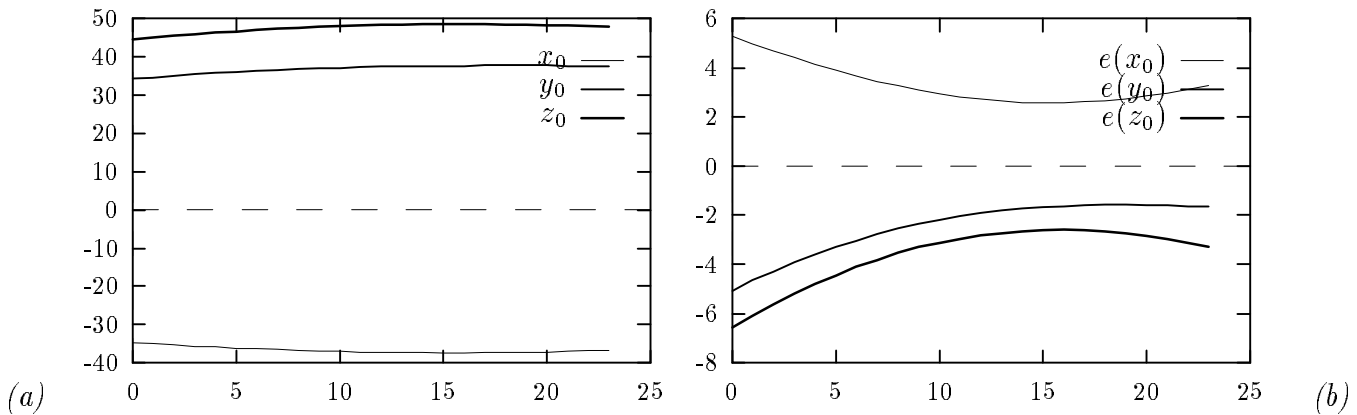


Figure 2.9 : *Reconstruction 3D d'un point par vision dynamique : (a) valeurs estimées à chaque itération des coordonnées 3D du point. (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*

Les erreurs obtenues sont relativement importantes pour l'ensemble des paramètres du point. Ces erreurs sont dues, comme on le verra par la suite, au phénomène de discrétisation qui apparaît lorsque le mouvement de la caméra est quelconque.

2.3.6.2 Exemple 2 : la sphère

Comme dans le cas précédent, la figure 2.10 illustre dans sa partie gauche la position initiale de la caméra par rapport à une sphère et dans sa partie gauche l'image correspondante.

En partant de cette position, on applique un mouvement quelconque à la caméra (le même que dans le cas du point). Les résultats obtenus sont décrits sur la figure 2.11, configurée comme la figure 2.9. On estime ici, à chaque itération, les coordonnées x_0, y_0, z_0 du centre de la sphère et son rayon r . Comme le cas du point, les estimations obtenues sur les paramètres de la sphère sont biaisées.

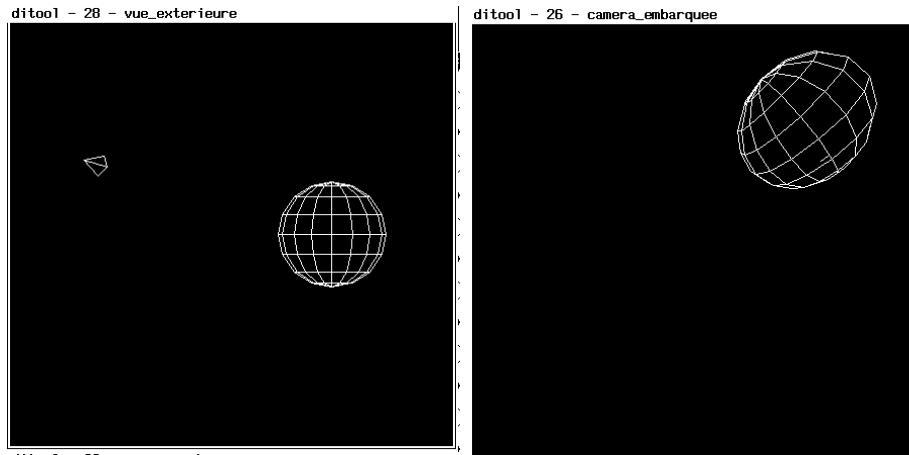


Figure 2.10 : Position entre la caméra et la sphère

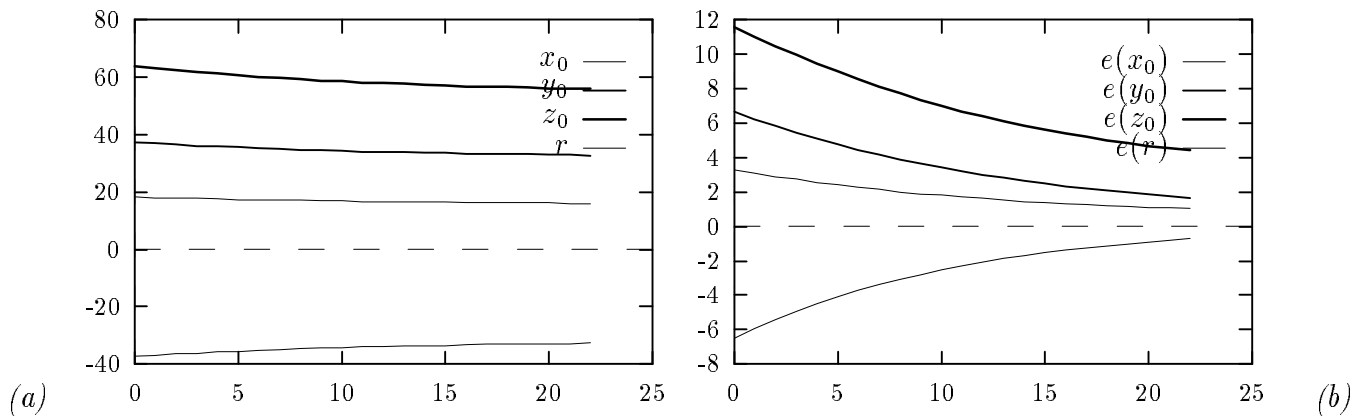


Figure 2.11 : Reconstruction 3D d'une sphère par vision dynamique : (a) valeurs estimées à chaque itération des coordonnées x_0, y_0, z_0 du centre de la sphère et de son rayon r . (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.

2.3.6.3 Exemple 3 : le cylindre

Nous montrons sur la figure 2.12, configurée comme la figure 2.8, la position initiale choisie entre la caméra et le cylindre.

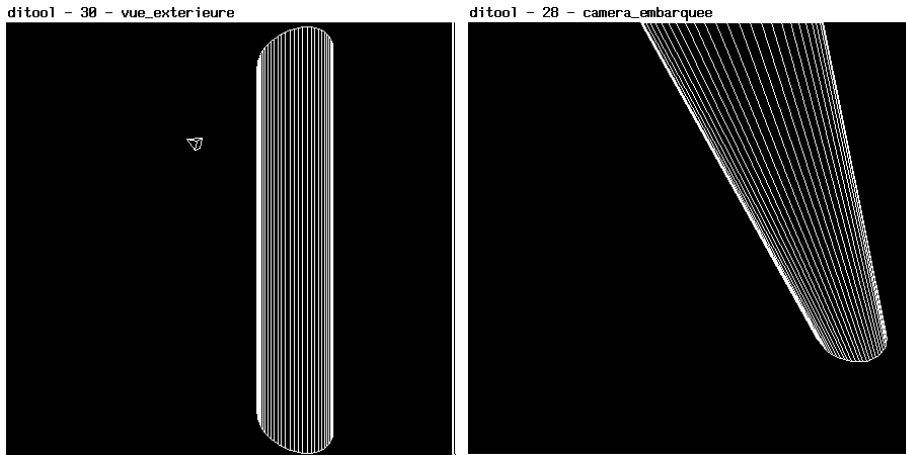


Figure 2.12 : *Position entre la caméra et le cylindre*

Comme pour les deux cas précédents, la caméra est ensuite animée d'un mouvement quelconque. Les paramètres que nous estimons ici sont les composantes (a, b, c) du vecteur directeur de l'axe du cylindre, son rayon r et les coordonnées (x_0, y_0, z_0) du point de son axe le plus proche du centre optique (voir paragraphe 2.3.5). Les résultats obtenus sont illustrés sur la figure 2.13 : à gauche, les différentes valeurs de a, b, c (en haut) et x_0, y_0, z_0, r (en bas) et, à droite, l'erreur entre ces valeurs estimées et les valeurs réelles. Notons que ces résultats ont été obtenus en utilisant la méthode basée sur les deux limbes. Des résultats similaires ont été obtenus en utilisant l'autre méthode (basée sur un seul limbe).

On observe ici encore les mêmes erreurs que dans les deux cas précédents. Notons que ce phénomène a été aussi observé pour les primitives droite et cercle.

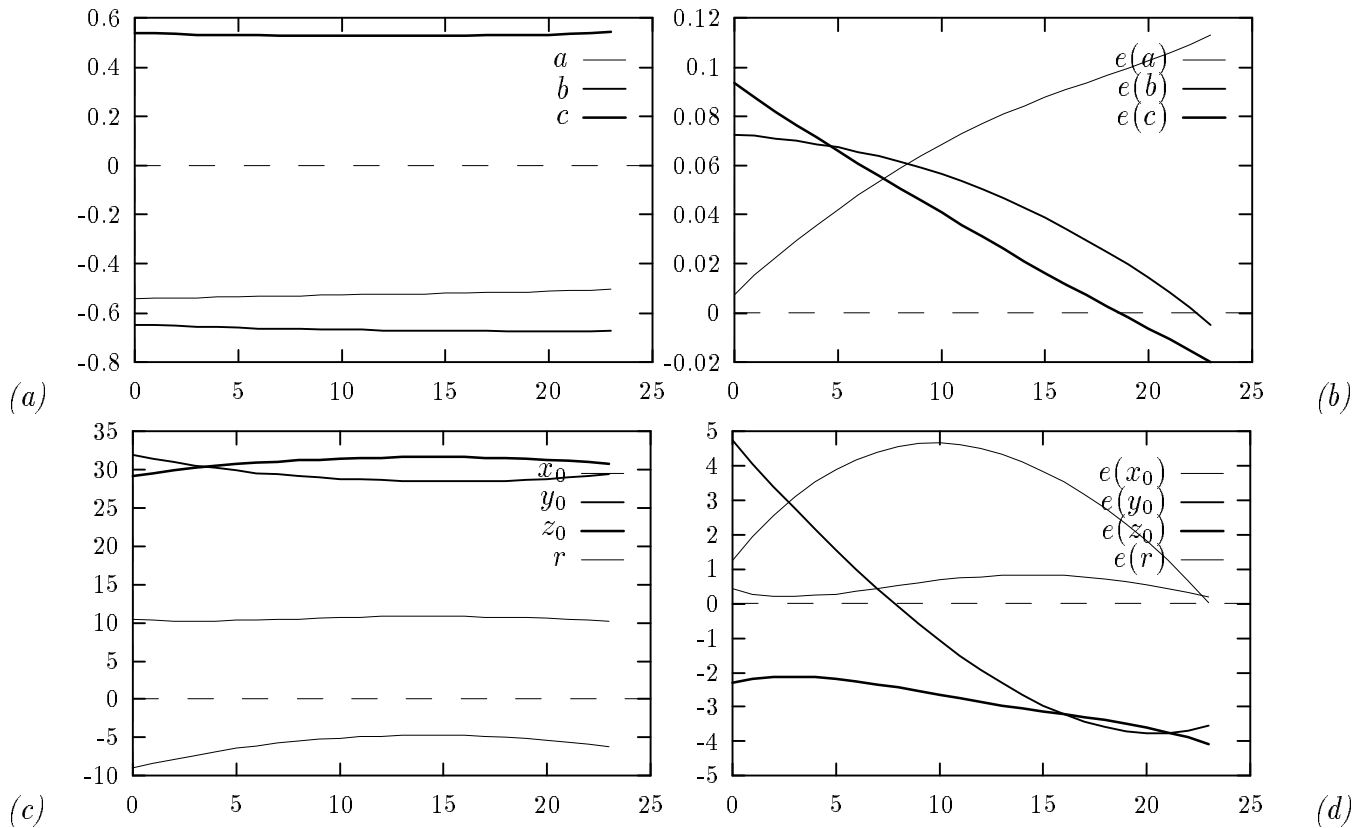


Figure 2.13 : *Reconstruction 3D d'un cylindre par vision dynamique : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles. (c) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r . (d) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*

2.4 Conclusion

Nous avons présenté dans ce chapitre une méthode générale de reconstruction des primitives géométriques paramétrables par vision dynamique. Dans notre démarche, nous avons généralisé les résultats déjà acquis concernant les primitives de type point et segment à des primitives plus complexes. Plus précisément, l'utilisation du tenseur d'interaction associé aux primitives de l'environnement a permis d'établir une méthode s'appliquant aisément aux primitives telles que cylindres, sphères, etc.

Néanmoins, malgré son élégance mathématique, notre approche s'avère inefficace pour reconstruire précisément les primitives de la scène. En effet, du fait de son aspect continu, elle est inévitablement confrontée aux erreurs de discrétisation comme en témoignent les résultats de simulation qui viennent d'être présentés. Elle souffre en outre, comme nous le verrons par la suite, de sa sensibilité au bruit. Pour améliorer la robustesse des solutions, des processus d'optimisation non linéaire [WEN92b], [VIA92] ou encore des processus de filtrage récursif [BOU89a], [VIA92], [CRO92] ont été introduits dans la littérature (voir le chapitre 1) aboutissant alors à des méthodes assez lourdes.

En ce qui nous concerne, nous adopterons une toute autre procédure pour faire face aux perturbations. Nous nous focaliserons sur le mouvement de la caméra. A notre sens, le mouvement de la caméra qui est à la base même de toutes les approches de reconstruction 3D par vision dynamique ne doit pas être quelconque. Il est clair que, comme en témoignent les résultats que nous venons de montrer, sans stratégies relatives au mouvement de la caméra la qualité des estimations est loin d'être satisfaisante.

Dans le chapitre suivant, le but sera donc de trouver des mouvements adéquats de la caméra qui permettront d'améliorer les résultats que nous avons obtenus [BOU92b], [CHA92], [BOU93c]. Ainsi, nous ne nous contenterons plus d'utiliser la vision dynamique qui nous paraît impuissante devant la complexité du problème de la structure à partir du mouvement. Nous lui préférons la vision active qui, elle, permet une interaction intelligente avec l'environnement afin d'opérer dans des conditions optimales.

Chapitre 3

Reconstruction 3D par vision active

Dans ce chapitre, nous faisons appel à la vision active pour améliorer la performance des méthodes présentées dans le chapitre précédent. Après avoir parcouru les principaux aspects de la vision active à travers une étude bibliographique, nous présentons son application à la reconstruction 3D de primitives géométriques. En effet, nous montrons que des mouvements adéquats de la caméra, engendrés par des lois de commande en boucle fermée sur les informations visuelles, permettent d'affiner de manière très significative la reconstruction 3D des primitives. Divers résultats de simulation complètent ce chapitre.

3.1 Introduction

La vision active est un concept récent défini comme un processus de perception intelligente de données. Plus précisément, la vision active consiste à élaborer des stratégies de contrôle et de modélisation de perception afin d'améliorer la connaissance de l'environnement. Les stratégies de contrôle consistent à rechercher des séquences de tâches qui minimiseraient une fonction de coût en recherchant le maximum d'informations pertinentes dans la scène à analyser.

Notons que Ballard préfère désigner la “vision active” par le terme “vision animée”, le terme précédent étant susceptible d'être confondu avec l'activité des capteurs qui est une notion complètement différente (voir chapitre 1).

La vision active se veut exploratoire tentant ainsi de simuler “tant bien que mal” la perception humaine. On ne regarde pas uniquement mais on observe, on cherche, on interroge. Pour cela, nos pupilles s’adaptent au niveau de l’illumination, nos yeux convergent ou divergent, nos têtes bougent pour avoir une meilleure vue de la zone d’intérêt. Cette faculté d’adaptation est cruciale pour mener à bien nos nombreuses tâches quotidiennes.

De nombreuses capacités très avantageuses caractérisent un système de vision actif par rapport à un système passif, par exemple, la possibilité de s’approcher des zones d’intérêt, de se focaliser sur une cible de la scène, d’aller observer une partie occultée du champ visuel, de lever l’ambiguïté sur certains aspects du monde perçu, etc. Ce type d’exploration est souvent plus efficace et en plus moins coûteux qu’une recherche algorithmique exhaustive sur une seule image dans laquelle on n’est même pas sûr qu’elle contienne l’objet d’intérêt.

Un exemple typique d’observateur actif est un système binoculaire dont les “yeux” peuvent effectuer des mouvements de rotation [BAL91], [DU 91], [VIE92], [MIL93]. Cette faculté le distingue ainsi des systèmes stéréoscopiques classiques souvent critiqués pour leur rigidité et leur manque d’adaptabilité (voir chapitre 1).

La principale motivation qui a entraîné l’émergence de premiers travaux en vision active est le fait que la vision passive s’est montrée très problématique. La plupart des problèmes de base de la vision passive sont très difficiles à résoudre parce que mal posés. Comme il n’existe souvent pas une solution unique au problème, celui-ci doit être régularisé en imposant des contraintes physiques supplémentaires. En effet, la plupart des problèmes mal posés sont non suffisamment contraints. L’idée de base des méthodes de régularisation est de restreindre l’espace des solutions admissibles en choisissant la fonction minimisant une fonctionnelle appropriée [POG85]. La régularisation est un paradigme très attrayant par son élégance mathématique. Cependant, son champ d’application est très restreint et s’avère incapable de couvrir toute la complexité de la vision. Cela est dû essentiellement au fait que cette approche prend en compte des contraintes beaucoup trop éloignées de la réalité. Par exemple, l’interpolation d’une surface est réalisée en lissant de façon trop prononcée les discontinuités de profondeur. Par ailleurs, il ne suffit pas de contraindre le problème pour qu’il n’ait qu’une solution unique. Cette solution peut être instable dans le sens où une petite erreur de mesure peut engendrer des résultats catastrophiques.

La vision active permet non seulement de passer des problèmes mal posés aux problèmes bien posés mais aussi des problèmes instables à des problèmes stables. Une étude particulièrement intéressante sur ce nouveau paradigme qu’est la vision

active et à laquelle se réfèrent quasiment la totalité des travaux dans ce domaine est proposée par Aloimonos *et al*, [ALO87]. D'autres travaux présentés par Aloimonos et ses étudiants [HUA91], [HER92] se focalisent aussi essentiellement sur la formulation mathématique des problèmes de stabilité, de linéarité et d'unicité des solutions.

A la différence de Aloimonos *et al* [ALO87] qui définissent et abordent la vision active d'un point de vue mathématique, Bajcsy [BAJ88] la définit comme une étude de stratégies de modélisation et de contrôle de la perception incluant la modélisation de capteurs, des objets, de l'environnement et des interactions entre eux pour la réalisation d'une tâche donnée. Des paramètres ainsi que les incertitudes associées doivent être définis. Ces paramètres peuvent alors être utilisés comme retour d'information afin de contrôler le système d'acquisition. Ce problème est loin d'être simple car la plupart de ces paramètres sont dépendants de la scène. Les incertitudes associées quant à elles dépendent de différents paramètres dont les paramètres de la caméra, les paramètres d'illumination, de réflectance, des algorithmes de traitement d'images (détection de contour, reconstruction 3D, etc) ainsi que de la tâche elle-même.

Selon Bajcsy, un système de vision actif est un processus appelé modèle global supervisant le contrôle et l'interaction de plusieurs processus ou modèles locaux. Ces modèles sont par exemple un module de calibrage, un algorithme de suivi de contours, etc. Ces différents processus individuels produisent des paramètres internes qui doivent varier dans une plage bien définie de valeurs possibles ainsi que les incertitudes associées. Cela permet au modèle global d'émettre des prédictions, notamment sur les interactions mutuelles des modèles locaux, et de combiner leurs différents résultats intermédiaires afin de guider le processus. La structure modulaire de ce système permet de simplifier relativement la complexité des tâches à réaliser. Enfin, ce système doit gérer aussi l'acquisition des informations. Un supplément d'informations n'est acquis qu'en cas de nécessité et en quantité strictement suffisante, l'optique étant non seulement d'opérer dans des conditions optimales, mais aussi en utilisant une quantité de ressources limitée.

Enfin, Bajcsy est l'une des premiers chercheurs à avoir défini la vision active et à avoir attiré l'attention de la communauté scientifique sur ce nouveau paradigme. Cependant, ses travaux restent très vagues et à notre connaissance aucune expérimentation effective n'a été rapportée dans la littérature à ce jour.

D'autres auteurs tels que Brown définissent un observateur actif comme un système capable de décider quel type d'information serait pertinent pour réaliser une

tâche donnée, quels types d’opérateurs utiliser, à quelle résolution et où les appliquer [BRO92]. Pour Tistarelli et Sandini, le terme “vision active” traduit le rôle fondamental du mouvement ou, de manière plus précise, l’interaction entre l’observateur avec l’environnement [TIS92]. Ce concept souligne l’intérêt de déterminer les propriétés de la scène à partir de l’évolution temporelle des caractéristiques de l’image. Le tableau 3.1 résume les différences fondamentales entre la vision passive standard ou paradigme de Marr [MAR82] et la vision active [BRO92].

Vision passive	Vision active
utilise tous les modules	utilise certains modules
traite toute l’image	traite certaines régions de l’image
maximum de détails	suffisamment de détails
construit une représentation d’abord	interroge d’abord
réponse à partir de la représentation	réponse à partir de la scène
ressources illimitées	ressources limitées

Tableau 3.1 : *Comparaison des approches active et passive.*

Enfin, la vision active permet également de réduire le temps de calcul pour extraire des informations pertinentes de l’image, voire opérer en temps réel [BAL91], par exemple la recherche se focalise sur des zones d’intérêt ou petites fenêtres (voir le chapitre 4 pour une bonne illustration de ce type de recherche).

3.2 Etat de l’art

Dans ce paragraphe, nous allons rapporter les contributions qui nous ont parues le plus significatives dans le domaine de la vision active, notamment celles qui touchent de près le problème qui nous intéresse tout particulièrement, à savoir la reconstruction 3D d’une scène. Vu l’apparition très récente du paradigme de la vision active, peu de travaux ont émergé dans la littérature. Combien même des investigations ont été faites dans ce domaine, les expérimentations en environnement réel restent très rares.

Dans ce qui suit nous allons aborder les aspects suivants de la vision active :

- le contrôle des paramètres oculaires;
- la tête binoculaire active;
- l’estimation du mouvement 2D par vision active;

- l'estimation du mouvement 3D par vision active;
- la reconstruction 3D par vision active.

Chaque point sera développé de manière relativement succincte. Le lecteur intéressé pourra consulter les références que nous proposons pour chaque point. Enfin, nous nous attarderons sur le dernier aspect qui constitue l'objectif principal de cette thèse.

3.2.1 Contrôle des paramètres oculaires

Comme le système humain de vision, un système de vision actif peut être doté d'un mécanisme de réflexes lui permettant d'avoir l'acuité visuelle la plus forte possible afin d'avoir une meilleure perception de l'environnement. Ces réflexes sont acquis en contrôlant les paramètres oculaires du diaphragme (ouverture, fermeture), du focus, de la vergence (convergence ou divergence des yeux d'un observateur binoculaire).

Ce mécanisme permet à l'observateur :

- d'accommoder l'œil (ou les yeux) pour maintenir une qualité satisfaisante de l'image en dépit des dégradations qui peuvent survenir au cours du processus d'acquisition.
- d'ajuster avec précision le regard de l'observateur sur une zone d'intérêt de l'environnement.
- de fournir une meilleure estimation de la position des objets de la scène.

Toujours dans l'optique d'avoir une acuité visuelle performante, le système peut aussi être doté d'un mécanisme lui permettant d'explorer la région d'intérêt (fovéa) avec une haute résolution tout en limitant la quantité d'informations acquises. C'est ce qu'on appelle la perception à champs variables.

Nous allons donc successivement aborder ces différentes facultés d'adaptation d'un observateur actif, à savoir le contrôle du focus, le contrôle du diaphragme et la perception à champs variables. Le contrôle de la vergence qui concerne uniquement l'observateur binoculaire sera abordé dans le paragraphe suivant.

3.2.1.1 Le contrôle du focus

Il est bien connu que le focus peut être contrôlé en utilisant le degré de netteté ou d'acuité du contraste de l'image acquise. Tout le problème réside dans le choix de la mesure de cette netteté. Crowley *et al.* contrôlent le focus de leur tête binoculaire active en utilisant comme mesure de netteté la somme des extrémas locaux des amplitudes du gradient de l'intensité lumineuse des pixels situés à l'intérieur de la région d'intérêt [CRO93]. Cette région est située par défaut au centre de l'image (fovéa). Ils recherchent alors la valeur qui maximise cette somme en jouant sur les paramètres oculaires du focus. Il faut assurer par la suite le maintien du focus sur un maximum local. Notons que cette mesure induit un plateau autour de la bonne valeur. Autrement dit, elle ne conduit pas à un pic franc, unique. Ce plateau sera d'autant plus large que le champ de vue sera large. Enfin, cette mesure présente l'avantage d'être simple et peu coûteuse.

3.2.1.2 Le contrôle du diaphragme

Comme le focus, le diaphragme d'un système de vision peut aussi être contrôlé afin d'ajuster le contraste de l'image. Une procédure très simple consiste à utiliser le niveau d'intensité lumineuse d'un seul pixel de l'image. Malheureusement, si le pixel considéré tombe sur une région sombre de la scène, cela peut conduire le système à ouvrir complètement le diaphragme saturant ainsi le reste de l'image.

Le contraste optimum est obtenu lorsque la dynamique du signal est plus importante. Crowley *et al.* considèrent chaque pixel de la région d'intérêt comme une variable aléatoire [CRO93]. Deux estimateurs sont définis : la variance du signal et l'amplitude du signal qui varie d'une valeur minimale de l'intensité lumineuse à une valeur maximale. Ils montrent alors que l'estimateur le plus robuste est celui qui maximise la variance des pixels dans la région d'intérêt. Cet estimateur a l'avantage de conduire à un pic unique et par conséquent à un résultat précis. Il fournit une compensation automatique entre le contraste et le focus. Enfin, il est stable pour différentes images de la même scène dans les mêmes conditions de luminosité et avec les mêmes paramètres oculaires.

3.2.1.3 La perception à champs variables

Dans le système humain de perception, les récepteurs de la rétine sont distribués avec une densité croissante en allant vers le centre du champ visuel. Cette petite région située aux alentours de l'axe optique est appelée fovéa. Cette distribution

non uniforme des récepteurs de la rétine explique la caractéristique de l'œil humain d'avoir une forte acuité visuelle locale tout en ayant un large champ visuel.

La perception à champs variables (échantillonnage non uniforme) tente de simuler ce phénomène. Le but est d'explorer l'environnement à haute résolution (avec le fovéa) sans encombrer le champ visuel avec des informations ou détails inutiles. En effet, la partie périphérique du champ visuel est perçue à basse résolution. On parle alors de vision fovéale (champ faible, haute résolution) et de vision périphérique (champ large, basse résolution). Ces deux possibilités de perception doivent être coordonnées et contrôlées par un processus capable de diriger le fovéa sur le point désiré de la scène (point de fixation) [TIS92].

La perception à champs variables, même si elle paraît plus difficile à gérer comparée à la perception classique, présente des avantages incontestables. Rien d'étonnant puisqu'on fait un pas de plus vers la perception humaine. Certes, on est encore bien loin du système de perception d'un simple insecte! Toujours est-il que bon nombre de tâches visuelles gagneraient aussi bien en précision qu'en fiabilité avec un tel système. En particulier, le suivi d'une cible mobile serait plus efficace en utilisant un système de ce type. En effet, cette tâche implique le calcul de petits déplacements aux alentours du centre de l'image (erreurs de poursuite). En utilisant la perception à champs variables, l'estimation des erreurs de poursuite sera effectuée avec le maximum de précision puisqu'elle se fera près du fovéa qui a la particularité de bénéficier de la plus haute résolution.

Enfin, il est clair que le développement de processus tels que ceux que nous venons de décrire (contrôle du focus, etc) s'avère indispensable pour la construction d'un système temps-réel de vision active.

3.2.2 Tête binoculaire active

3.2.2.1 Le point de fixation

Plusieurs travaux ont montré récemment que les problèmes devenaient plus simples lorsque l'observateur (monoculaire ou binoculaire) était capable de fixer un point choisi de l'environnement [ALO87], [BAN86], [SAN86], [SAN90], [LI 91], [PAH92] au cours de son déplacement. Le mouvement de l'observateur doit être contrôlé de manière à ce que le point de l'environnement apparaisse en permanence immobile centré dans l'image, autrement dit que le point soit constamment sur l'axe optique. Ce point est communément appelé *point de fixation*. Dans le cas d'un observateur

binoculaire, les axes optiques des deux caméras doivent être constamment dirigés vers le point choisi de l'environnement.

Dans [BAL91], Ballard met l'accent sur l'importance du choix d'un repère lié à la scène (repère objet) dans lequel doivent être exprimés les calculs, avis complètement divergent avec la théorie de Marr qui argumente que les calculs doivent être exprimés dans un repère lié à l'observateur [MAR82]. En effet, la capacité de contrôler le regard de la caméra et particulièrement de fixer des cibles dans l'environnement au cours du mouvement du robot, permet à celui-ci de choisir un repère de référence lié à l'environnement (repère objet). Un choix judicieux consiste en un repère centré au point de fixation, ses axes étant parallèles à ceux du repère lié à l'observateur (repère caméra). Les calculs sont alors largement simplifiés. En outre, ce repère présente l'avantage d'être invariant par rapport au mouvement de l'observateur.

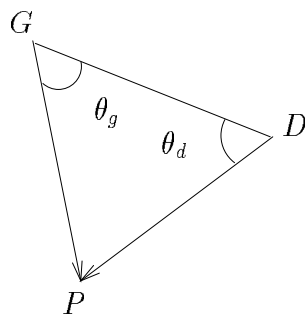
Un autre avantage du point de fixation en plus des simplifications qu'il apporte dans les différents problèmes abordés (poursuite de cible, navigation, reconstruction 3D, etc) réside dans le fait que les mesures sont acquises près de l'axe optique, à l'endroit où les distorsions optiques sont le moins ressenties.

En ce qui nous concerne, nous montrerons que la reconstruction 3D d'un point de la scène est optimale quand justement celui-ci apparaît constamment centré dans l'image [BOU92b]. Cela confirme les propriétés du point de fixation qui s'est avéré très avantageux dans différentes applications notamment la reconstruction 3D [SAN86], [SAN90], la détermination du mouvement 3D [ALO87], [BAN86] ou la manipulation d'objets [ZHE91], [LI 91].

3.2.2.2 Le contrôle de la vergence

Dans [BAL91], [VIE92], [DU 91], [PAH92], [CRO93], [MIL93] les auteurs insistent sur l'importance de la binocularité dans un système de vision active. Cette capacité doit être appuyée par un processus puissant de contrôle du regard. C'est ce qu'on appelle la vergence active (convergence ou divergence des deux "yeux"). Quand le point 3D d'intérêt est spécifié, le système détermine les angles de vergence θ_a, θ_g (voir figure 3.1) des deux caméras qui permettent d'amener les deux axes optiques à s'intersecter au point de fixation spécifié. Une mesure de convergence robuste utilisant un filtre bien spécifique est décrite dans [CRO93].

Sans la vergence, de très larges disparités peuvent être obtenues. D'un autre côté, si la vergence peut être contrôlée, ces disparités peuvent être maintenues à une grandeur satisfaisante autrement dit, assez petites pour permettre une mise en correspondance aisée mais aussi assez grandes pour obtenir des résultats suffisam-

Figure 3.1 : *Observateur binoculaire actif.*

ment précis. Le contrôle du regard est donc un moyen puissant pour contourner (du moins en partie) les problèmes de la mise en correspondance des primitives bidimensionnelles.

3.2.3 Estimation du mouvement 2D par vision active

Un travail allant dans ce sens a été réalisé par Xie dans son travail de thèse [XIE89]. Rappelons que son objectif principal était de recouvrer la structure 3D d'objets polyédriques à l'aide d'une caméra mobile commandable. Il s'est donc penché sur la question de trouver des trajectoires optimales de la caméra en vue de mieux reconstruire, question tout à fait légitime compte tenu du mal qu'ont en général les approches de reconstruction 3D à atteindre une précision acceptable. Nous nous poserons nous-même la même question dans le cadre de notre travail en l'approfondissant grandement et pour une catégorie moins restreinte d'objets que la classe des objets polyédriques.

Xie utilise une loi de commande qui permet de générer une trajectoire optimale de la caméra le long de laquelle la reconstruction 3D sera la plus précise. L'idée est de contrôler localement le mouvement de la caméra qui permet d'assurer l'observation d'un mouvement 2D apparent optimal. Nous verrons que nous aussi ferons appel à un schéma de commande en boucle fermée pour assurer la génération automatique des mouvements optimaux de la caméra.

Pour un segment 2D, l'estimation de son mouvement apparent sera optimal si le champ de déplacements perçu dans le plan image est perpendiculaire à la direction du segment. Caractériser de manière globale le champ de déplacements correspon-

nant à un ensemble de segments paraît beaucoup moins évident. Le mouvement apparent est *globalement optimal* si le champ de déplacements perçu est dans une direction perpendiculaire à un sous-ensemble le plus grand possible des segments 2D estimés. Ce champ étant impossible à caractériser théoriquement, le choix de l'auteur a porté sur la projection du déplacement du centre de gravité de la scène 3D. Ce vecteur de déplacements doit être perpendiculaire à la direction moyenne pondérée de l'ensemble des segments 2D et sa norme doit être égale à la moyenne pondérée des normes souhaitées des vecteurs de déplacement pour l'ensemble des segments 2D.

Ayant défini le mouvement apparent optimal pour la reconstruction 3D d'une scène polyédrique, il est alors toujours possible de trouver un mouvement 3D qui assure l'observation de celui-ci, ce mouvement local étant généré par une loi de commande en boucle fermée. Cette approche a été validée sur des scènes synthétiques relativement simples. Les résultats obtenus sont intéressants. Il reste que ces résultats ne peuvent pas vraiment refléter le comportement du système en environnement réel dans lequel interviennent énormément de facteurs qu'il est impossible de simuler.

Enfin, ce travail a mis en évidence l'intérêt d'obtenir des lignes épipolaires adaptatives en utilisant une caméra mobile commandable. Il démontre ainsi l'efficacité et la souplesse de cette approche monoculaire par rapport à la stéréovision dans laquelle les lignes épipolaires sont fixes ce qui constitue une limitation de cette dernière approche. En effet, une partie non négligeable des segments de la scène peut être pénalisée par ce manque d'adaptabilité (segments parallèles aux lignes épipolaires).

3.2.4 Estimation du mouvement 3D par vision active

Le problème de la structure à partir du mouvement est bien connu pour son instabilité (voir chapitre 1). Aloimonos, Bandopadhyay *et al.*, [ALO87], [BAN86], abordent ce problème crucial en adoptant une approche continue. Plus exactement, leur but consiste à recouvrer le mouvement 3D d'un observateur se déplaçant dans un environnement statique. Dans leur formulation du problème, ils supposent que le mouvement apparent 2D est la projection dans l'image du mouvement 3D de la scène. Ils montrent alors qu'un observateur actif résoud le problème de la structure à partir du mouvement de manière plus efficace :

- un observateur monoculaire actif ramène la dimension de l'espace des inconnues (paramètres du mouvement de l'observateur) de 5 à 4 sans toutefois s'affranchir des non linéarités.

- un observateur binoculaire actif simplifie énormément les contraintes liées au mouvement et permet d'aboutir à une solution simple en s'affranchissant de calculs itératifs prohibitifs et souvent peu fiables.

Nous allons à présent décrire plus en détails ces deux méthodes.

3.2.4.1 Cas d'un observateur monoculaire

Si l'observateur monoculaire peut discerner et effectuer la poursuite d'une caractéristique pertinente de la surface observée (un point ici), le problème est alors simplifié. En supposant que la vitesse apparente du point caractéristique choisi est connue avec précision, alors la dimension des inconnues des équations du champ 2D de vitesses apparentes est réduite de 1 sans augmenter la complexité des équations. Un résultat similaire est obtenu lorsque l'observateur est capable de suivre le point caractéristique tel qu'il apparaisse immobile centré dans l'image (point de fixation).

Il existe deux types de mouvement de poursuite : un mouvement par saccades et un mouvement continu. C'est ce dernier qui est utilisé ici. Deux degrés de liberté (rotations Ω_y et Ω_x appelées aussi *lacet* ("pan") et *tangage* ("tilt") de l'observateur sont utilisés pour réaliser cette tâche de poursuite. Ainsi, le champ de vitesses apparentes va présenter une singularité à l'origine de l'image où sa valeur est proche de zéro. L'orientation de la caméra est constamment ajustée en fonction du mouvement apparent 2D observé. C'est un système en boucle fermée.

3.2.4.2 Cas d'un observateur binoculaire

La même stratégie est aussi utilisée dans le cas d'un observateur binoculaire afin de simplifier le problème de l'estimation du mouvement 3D. Les axes optiques des deux caméras du système binoculaire sont constamment dirigés vers le point de fixation choisi.

Le mouvement de l'observateur binoculaire est supposé rectiligne uniforme et son mouvement de poursuite consiste en trois rotations indépendantes : une rotation autour de la ligne de base reliant les deux caméras, les deux autres étant liées respectivement à la caméra gauche et à la caméra droite. L'orientation des deux caméras est ainsi constamment ajustée afin de maintenir la convergence des deux axes sur le point de fixation. Cette capacité de l'observateur simplifie de manière très significative les contraintes liées au mouvement et permet d'aboutir à une solution simple.

La résolution du système d'équations obtenu ne requiert pas la mise en œuvre d'une procédure itérative. En effet, une seule itération suffit pour aboutir à au plus 4 solutions possibles pour les paramètres de mouvement. Les seules observations sont $(\theta_d, \theta_g, \dot{\theta}_d, \dot{\theta}_g)$ (voir figure 3.1), la mesure du mouvement apparent 2D ne s'avérant pas nécessaire. C'est une propriété qui ne manque pas d'intérêts vu les difficultés pour estimer correctement le champ 2D de vitesses apparentes. En fait, le mouvement apparent est quand même sollicité mais juste pour lever les ambiguïtés sur les différentes interprétations du module de suivi. Il n'a donc effectivement pas d'influence quantitative directe sur les résultats de cette approche.

Les auteurs n'ayant pas illustré leur approche par une expérimentation réelle, la précision des mesures sur les quantités $(\theta_d, \theta_g, \dot{\theta}_d, \dot{\theta}_g)$ et surtout son influence sur le calcul des paramètres du mouvement 3D restent à évaluer.

3.2.4.3 Discussion

Les travaux d'Aloimonos, Bandopadhyay *et al* [ALO87], [BAN86] qui viennent d'être présentés mettent l'accent sur les potentialités très prometteuses de la vision active en abordant un des problèmes critiques de la vision. Il aurait tout de même été souhaitable que leurs théories soient appuyées par des expérimentations réelles. Et l'on sait bien combien la vision ou la robotique en général se montrent très problématiques en pratique! La précision et la stabilité de ces solutions théoriques ne peuvent notamment convaincre qu'en faisant leur preuves dans le cadre d'une tâche réelle.

Enfin, les auteurs ne se sont pas penchés sur la question de savoir s'il existe des trajectoires de l'observateur qui soient plus bénéfiques que d'autres.

3.2.5 Reconstruction 3D par vision active

Sandini et Tistarelli dans [SAN90] proposent une approche active pour déterminer la carte des profondeurs d'un environnement statique perçu par une caméra en mouvement. La stratégie de déplacement de la caméra qu'ils ont adoptée est similaire à celle d'Aloimonos, Bandopadhyay *et al*. [ALO87], [BAN86], autrement dit, la caméra est contrôlée de manière à stabiliser le point de fixation au cours de son déplacement. Le mouvement ainsi contraint permet comme on l'a vu précédemment de déterminer de manière plus aisée les paramètres du mouvement de la caméra.

La détermination de la carte des profondeurs se fait en deux étapes :

- (1) la mise en correspondance de primitives points entre les images successives pour le calcul du mouvement 2D de ces points.
- (2) la détermination de la profondeur des points de la scène en utilisant le mouvement 2D ainsi que les paramètres du mouvement de la caméra précédemment calculés.

Le champ de vitesses apparentes est déterminé en utilisant une approche *mixte* qui optimise d'une part, la mise en correspondance des points d'image en image et d'autre part, la détermination des paramètres de mouvement ainsi que la profondeur des points. En effet, une procédure locale permet d'apparier les points avec efficacité en considérant des déplacements faibles entre les images successives. Une fois qu'un segment de trajectoire *suffisamment* long a été engendré par l'observateur, une procédure globale consiste alors à déterminer les paramètres du mouvement de la caméra en ne considérant que la première et la dernière image du segment. Enfin, la profondeur des points de la scène est déterminée en effectuant une triangulation sur les points de la première et de la dernière image de la séquence.

Cette approche a été validée sur des images réelles. Il reste que l'évaluation de ces résultats comparés à d'autres stratégies passives n'est pas évidente. Une étude comparative entre une approche passive et une approche active destinées à la même tâche et opérant dans des conditions similaires serait intéressante pour mettre effectivement en valeur l'apport de la vision active. C'est justement la démarche que nous avons suivie pour montrer les améliorations très significatives apportées par une approche de type vision active par rapport à une approche passive dans le domaine de la reconstruction 3D.

Un autre algorithme robuste d'estimation de la carte des profondeurs est aussi proposé par Sandini et Tistarelli dans [SAN86]. Cet algorithme intègre les informations visuelles issues de la stéréovision et du mouvement en se basant sur une stratégie de déplacement des caméras pouvant apporter des informations fiables dans les conditions où l'analyse stéréoscopique est mise en échec et vice versa. En effet, la géométrie du système stéréoscopique ne permet pas d'acquérir des informations de profondeur fiables pour les contours dont l'orientation dans le plan image est proche de celle des épipolaires. D'un autre côté le même problème se pose dans le cas de la vision dynamique pour les contours dont l'orientation dans le plan image est proche de la projection dans l'image de l'axe de la translation (mouvement de glissement du segment de contour sur lui même).

Pour optimiser la contribution mutuelle des deux systèmes, la stratégie adoptée consiste à contrôler le mouvement de la caméra tel que le point de fixation reste immobile (stabilisation) le long de la séquence d'images, la trajectoire du mouvement étant elle-même contrainte sur un plan vertical (la paire stéréoscopique se trouvant sur un plan horizontal).

Pour déterminer complètement le champ de vitesses apparentes issu d'une des deux caméras, les contraintes imposées sur le mouvement de la caméra sont utilisées. La composante de vitesse en rotation de la caméra est obtenue en mesurant l'angle de rotation de la caméra autour de l'axe x . En utilisant, sur deux images successives, cet angle et la distance de la caméra au point de fixation, distance obtenue par stéréovision, la direction de la translation peut être déterminée. En combinant celles-ci avec la composante de vitesse perpendiculaire au contour, un système linéaire est obtenu et conduit à l'interprétation complète du champ de vitesses apparentes. Le champ de vitesses apparentes étant déterminé, la carte des profondeurs est alors obtenue. Une seconde estimation de cette carte est obtenue parallèlement par stéréovision.

La procédure d'intégration consiste tout simplement à faire un moyennage sur les deux cartes de profondeur obtenues. Des résultats sur des images réelles illustrent cette approche originale. Malheureusement il est relativement difficile encore une fois d'évaluer visuellement l'amélioration obtenue par cette stratégie active par rapport aux méthodes classiques.

Malgré ses atouts incontestables, la vision active n'est pas en mesure à ce jour d'affronter les problèmes du monde réel dans toute leur complexité. D'un autre côté, si la vision active est intimement liée à la tâche bien spécifique à accomplir, le problème peut être largement simplifié. Il est inutile de s'acharner à calculer avec précision la structure 3D d'un chemin si le but est juste de le traverser ou d'en éviter les obstacles. Dans ce cas là, des mesures grossières, voire quelques paramètres qualitatifs, suffisent amplement.

Dans beaucoup d'applications, en robotique notamment, on ne maîtrise pas encore assez quel type de mesure répond le mieux à une application donnée. Aloimonos, [ALO90], s'est penché de près à cette question en soulignant les avantages de la vision qualitative active liée à un but bien précis. Le principe de base de cette approche consiste à dégager *des solutions générales à des problèmes spécifiques* se distinguant ainsi de la démarche classique qui consiste à trouver *des solutions à des problèmes généraux*. Cette dernière est critiquée pour son manque d'efficacité flagrant lors de ses confrontations à des problèmes réels.

Aloimonos propose donc de bien poser le problème et de trouver une solution adaptée uniquement à ce problème. C'est cette démarche qui est suivie dans le travail qu'il présente avec Huang dans lequel ils appréhendent le problème de l'estimation de la structure à partir du mouvement [HUA91]. Leur approche est non seulement active mais aussi intentionnelle ("purposive vision") dans le sens où ils recherchent une solution destinée juste à ce problème bien précis.

Ils montrent que ce problème peut être résolu en évitant l'étape intermédiaire de l'estimation du mouvement 3D et celle du mouvement apparent 2D. Seules les dérivées spatiales et temporelles de la fonction intensité lumineuse sont nécessaires. Celles-ci définissent les composantes perpendiculaires au contour des vitesses apparentes. Il est clair que l'estimation de ces dernières s'avère beaucoup plus aisée que celle du champ dense 2D de vitesses apparentes. Naturellement, il ne faut pas s'attendre, sans un champ dense de vitesses apparentes, à déterminer une carte de profondeur complète. Ce manque d'informations est compensé en partie par l'utilisation d'un observateur actif. Par ailleurs, une carte de profondeur complète n'est pas toujours requise, en particulier dans ce qui suit où le but est juste de localiser les objets de la scène les uns par rapport aux autres.

Les auteurs abordent le problème en commençant par le cas le plus simple d'un objet statique jusqu'au cas le plus complexe d'un objet animé d'un mouvement quelconque. Dans tous les cas, le mouvement de l'observateur est une translation le long de l'axe optique. Cela simplifie largement les équations mais on verra que ce n'est pas un mouvement optimal pour recouvrer la structure des points de la scène.

- **Objet statique** : il suffit juste de connaître le sens de déplacement de la caméra pour estimer les profondeurs relatives des objets de la scène.
- **Objet animé d'une translation parallèle au plan image** : une information partielle du mouvement est nécessaire à savoir, la direction de cette translation. La profondeur peut alors être estimée (à un facteur d'échelle près) mais juste pour une catégorie restreinte de points : ceux dont la composante normale (au contour) de la vitesse apparente correspondante est perpendiculaire à la direction du mouvement de translation de l'objet.
- **Objet animé d'une translation quelconque** : un déplacement supplémentaire de l'observateur est nécessaire. Le mouvement de l'objet est supposé constant et le signe de l'accélération de la caméra est supposé connu. La profondeur peut alors être estimée pour la même catégorie restreinte des points de l'objet que dans le cas précédent.

- Objet animé d'un mouvement quelconque : il suffit de considérer un repère lié au point considéré pour éliminer l'effet de la rotation et se ramener au cas précédent.

Enfin, dans tous les cas les expressions obtenues sont très simples et l'erreur relative sur la profondeur est une fonction linéaire des erreurs sur les composantes perpendiculaires au contour des vitesses apparentes. Cette approche active est donc potentiellement robuste au bruit. Des résultats de simulation et réels illustrent cette approche. Le but étant de localiser les objets (susceptibles d'être mobiles) de la scène les uns par rapport aux autres, l'approche donne des résultats tout à fait satisfaisants. Enfin, soulignons que la contrainte utilisée est une contrainte assez forte et ne peut satisfaire toutes les catégories d'objets qui peuvent se présenter dans la scène.

3.3 Conclusion

Les quelques travaux que nous venons de présenter démontrent que beaucoup de problèmes connus pour leur grande complexité en vision dynamique (mise en correspondance de primitives 2D, reconstruction 3D, etc) deviennent plus simples à résoudre et/ou plus robustes grâce à la vision active.

Les principales motivations des groupes de recherche ayant travaillé sur ce nouveau paradigme étaient de simuler des mécanismes biologiques. Des processus tels que le contrôle du focus, de la vergence,.. permettent d'améliorer l'acuité visuelle de l'observateur. D'autres motivations d'ordre mathématique tentent de transformer les problèmes mal posés en problèmes bien posés (stabilité, linéarité, unicité des solutions). Enfin, les motivations englobent tous les moyens permettant d'améliorer la perception de l'environnement, soit d'appréhender les problèmes de la vision avec plus d'efficacité.

Les travaux qui ont été rapportés dans la littérature restent malheureusement souvent au stade théorique. Des expérimentations réelles sont très rares. Ces travaux gagneraient à être plus convaincants quant à l'apport de la vision active par rapport à la vision dynamique si des expérimentations réelles venaient appuyer leurs concepts.

D'autre part, les capacités incontestables de la vision active sont sous-exploitées. Beaucoup d'investigations restent à faire pour explorer toutes les possibilités d'adaptation de la vision active aux multiples tâches qui pourraient lui être attribuées.

En ce qui nous concerne, nous avons adopté une stratégie qui sera avant tout très démonstrative, notre objectif étant de mettre en valeur les avantages de la vision active par rapport à la vision dynamique dans le domaine bien spécifique de la reconstruction 3D. Beaucoup d'expérimentations tant en simulation qu'en environnement réel illustreront notre approche optimale de reconstruction 3D. A notre connaissance, il n'existe pas de travaux dans la littérature qui aient effectué une étude comparative *objective* entre les deux approches dans le domaine de la reconstruction 3D, ni même d'ailleurs dans les autres domaines.

3.4 Reconstruction 3D de primitives géométriques par vision active

Nous désirons appliquer la vision active au problème de la reconstruction 3D de primitives géométriques paramétrables en nous appuyant sur la méthode de reconstruction par vision dynamique décrite dans le chapitre précédent.

Notre objectif ici est de déterminer des positions et mouvements adéquats de la caméra qui permettent une meilleure estimation de la structure spatiale des primitives. La qualité de l'estimation à partir d'une séquence d'images est en effet très sensible aux mouvements successifs de la caméra [ESP87]. Ce problème sera abordé sous les deux aspects suivants :

- (1) la suppression des effets des erreurs de discrétisation observés dans les résultats précédents ;
- (2) la minimisation des effets des erreurs de mesure : les erreurs de mesure sont inhérentes à tout algorithme de traitement d'images et, de plus, les imprécisions sur la mesure du mouvement de la caméra sont souvent importantes. Nous montrerons dans le paragraphe 3.4.3 qu'il existe des configurations et mouvements de la caméra qui permettent une reconstruction plus robuste des primitives.

3.4.1 Suppression des erreurs de discrétisation

La méthode de reconstruction 3D par vision dynamique que nous avons décrite dans le chapitre précédent est notamment basée sur la mesure de $\dot{\underline{P}}$, la vitesse des paramètres représentant la primitive considérée dans l'image. Rappelons en effet que les paramètres de la surface des limbes sont de la forme (voir (2.9)) :

$$\underline{p}_0 = \underline{p}_0(\underline{P}, \dot{\underline{P}}, T) \quad (3.1)$$

La mesure de $\dot{\underline{P}}$ est impossible à effectuer exactement puisque les acquisitions d'images sont effectuées à des instants discrets. Seule $\Delta\underline{P}$, la variation des valeurs de \underline{P} entre deux ou plusieurs images, peut être déterminée. L'utilisation de $\frac{\Delta\underline{P}}{\Delta t}$ dans (3.1) à la place de $\dot{\underline{P}}$ peut entraîner des erreurs sur l'estimation des paramètres \underline{p}_0 . Celles-ci se répercutent bien évidemment sur l'estimation des paramètres \underline{p} représentant la primitive dans la scène, d'où le biais observé dans les résultats du chapitre précédent.

Nous allons donc déterminer les contraintes à fixer sur le mouvement de la caméra afin de supprimer ce biais. En effet, si l'on peut assurer que :

$$\dot{\underline{P}} = \frac{\Delta\underline{P}}{\Delta t}, \quad \forall t \quad (3.2)$$

la discrétisation effectuée n'aura aucun effet sur les algorithmes d'estimation. Cette condition est satisfaite si et seulement si :

$$\ddot{\underline{P}} = \dots = \underline{P}^{[n]} = 0, \quad \forall t \quad (3.3)$$

Revenons à présent à l'expression de $\dot{\underline{P}}$. On a d'après (2.7) :

$$\dot{\underline{P}} = f(\underline{P}, \underline{p}_0, T) \quad (3.4)$$

d'où :

$$\ddot{\underline{P}} = \frac{\partial f}{\partial \underline{P}} \dot{\underline{P}} + \frac{\partial f}{\partial \underline{p}_0} \dot{\underline{p}}_0 + \frac{\partial f}{\partial T} \dot{T} \quad (3.5)$$

On peut considérer que l'accélération \dot{T} est nulle en choisissant pour T la vitesse moyenne de la caméra, calculée à partir de la mesure des positions de la caméra à chaque acquisition d'image. Une condition générale et suffisante (mais non forcément nécessaire) pour respecter la condition (3.3) est donc de contraindre les mouvements de la caméra de telle sorte que :

$$\dot{\underline{P}} = \dot{\underline{p}}_0 = 0, \quad \forall t \quad (3.6)$$

Cette condition signifie que l'image et la surface des limbes de la primitive (ou plus précisément les paramètres les représentant) doivent rester constantes le long de la trajectoire de la caméra. Pour le cas d'une sphère qui apparaît sous la forme d'un cercle centré dans l'image par exemple, on peut aisément montrer que cette condition signifie que la caméra doit se déplacer sur une sphère concentrique tout en pointant sur le centre de la sphère. Nous verrons dans le paragraphe 3.4.5 comment il est possible, en utilisant l'approche asservissement visuel, d'établir des lois de commande en boucle fermée sur la vision générant automatiquement les mouvements de la caméra vérifiant de telles contraintes.

• **Remarques :**

- (1) La condition (3.6) que nous avons établie n'est pas nécessaire et suffisante. En effet, pour chaque primitive, il existe des configurations telles que :

$$\frac{\partial f}{\partial \underline{P}} \dot{\underline{P}} + \frac{\partial f}{\partial \underline{p}_0} \dot{\underline{p}}_0 = 0 \text{ avec } \dot{\underline{P}} \neq 0 \text{ et } \dot{\underline{p}}_0 \neq 0 \quad (3.7)$$

Pour le cas du point, on rappelle qu'on a :

$$\dot{z} = -V_z - zY\Omega_x + zX\Omega_y \quad (3.8)$$

et on obtient en différenciant (2.11) :

$$\begin{cases} \ddot{X} = \frac{\dot{z}}{z^2}(V_x - XV_z) + \dot{X}\left(\frac{V_z}{z} + Y\Omega_x - 2X\Omega_y\right) + \dot{Y}(X\Omega_x + \Omega_z) \\ \ddot{Y} = \frac{\dot{z}}{z^2}(V_y - YV_z) + \dot{X}(-Y\Omega_y - \Omega_z) + \dot{Y}\left(\frac{V_z}{z} + 2Y\Omega_x - X\Omega_y\right) \end{cases} \quad (3.9)$$

\ddot{X} et \ddot{Y} s'annulent par exemple si la caméra n'est animée que de mouvements de translation ($\Omega_x = \Omega_y = \Omega_z = 0$). On a alors $\dot{z} = -V_z \neq 0$ si $V_z \neq 0$ et $\dot{X} = (-V_x + XV_z)/z$, $\dot{Y} = (-V_y + YV_z)/z$ qui ne peuvent s'annuler simultanément pour permettre la reconstruction du point. D'autres cas particuliers ont également pu être déterminés. La recherche de l'ensemble de ces configurations paraît cependant peu intéressante puisque particulière à chaque primitive. On préfère donc se limiter à la condition (3.6) qui a le mérite d'être valide quelle que soit la primitive considérée.

- (2) On peut également remarquer que la condition $\dot{\underline{P}} = 0$ entraîne $\dot{\underline{p}}_0 = 0$ pour les primitives de type sphère, cylindre (en observant ses deux limbes) et cercle (excepté le cas particulier où l'image du cercle est un cercle centré). Pour ces primitives, la condition $\dot{\underline{P}} = 0$ est donc suffisante pour supprimer les erreurs de discrétisation. Cela n'est pas le cas, par contre, pour les primitives de type point et droite dont la dimension est inférieure.

3.4.2 Résultats de simulation

3.4.2.1 Exemple 1 : le point

La figure 3.2 configurée comme la figure 2.9 montre les résultats de simulation obtenus en appliquant cette stratégie. Le processus de reconstruction a été déclenché à partir de la même position initiale de la caméra (voir figure 2.8). On peut observer que les erreurs de discrétisation disparaissent complètement après les quelques premières itérations nécessaires à la convergence de la commande.

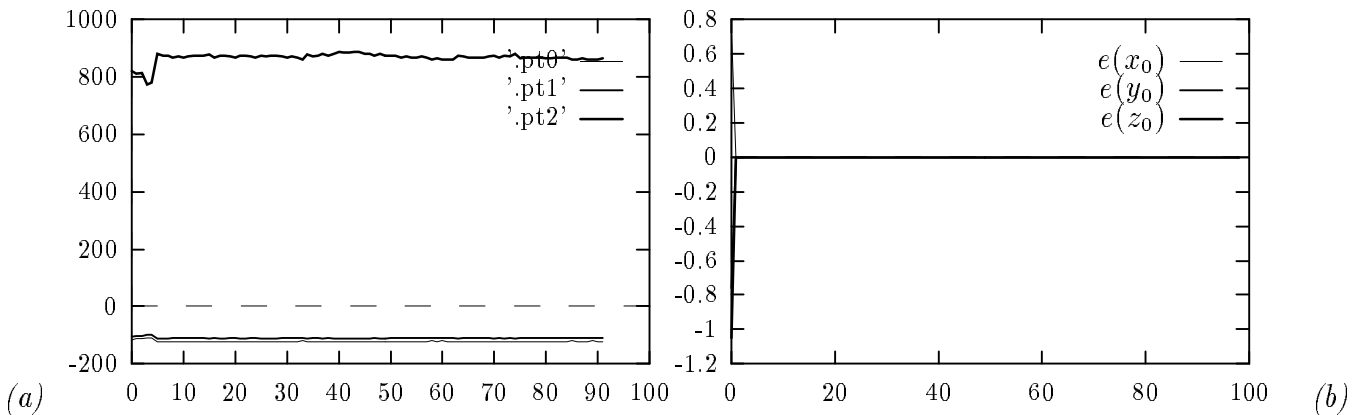


Figure 3.2 : *Reconstruction 3D d'un point en supprimant l'erreur de discrétisation : (a) valeurs estimées à chaque itération des coordonnées 3D du point. (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*
(sans bruit)

Si l'on introduit artificiellement du bruit sur les mesures image (bruit uniforme de 3 pixels pour des images 512×512) et sur la vitesse de la caméra (bruit uniforme de 2.5% sur chaque composante) pour simuler une mauvaise calibration et des erreurs de mesure sur la position de la caméra et les informations extraites des images, on obtient les résultats présentés sur la figure 3.3.

3.4.2.2 Exemple 2 : la sphère

Les résultats correspondant à la reconstruction d'une sphère en adoptant cette stratégie sont présentés sur la figure 3.4 configurée comme la figure 2.11.

En introduisant du bruit artificiel (le même que dans le cas du point), on obtient les résultats présentés sur la figure 3.5.

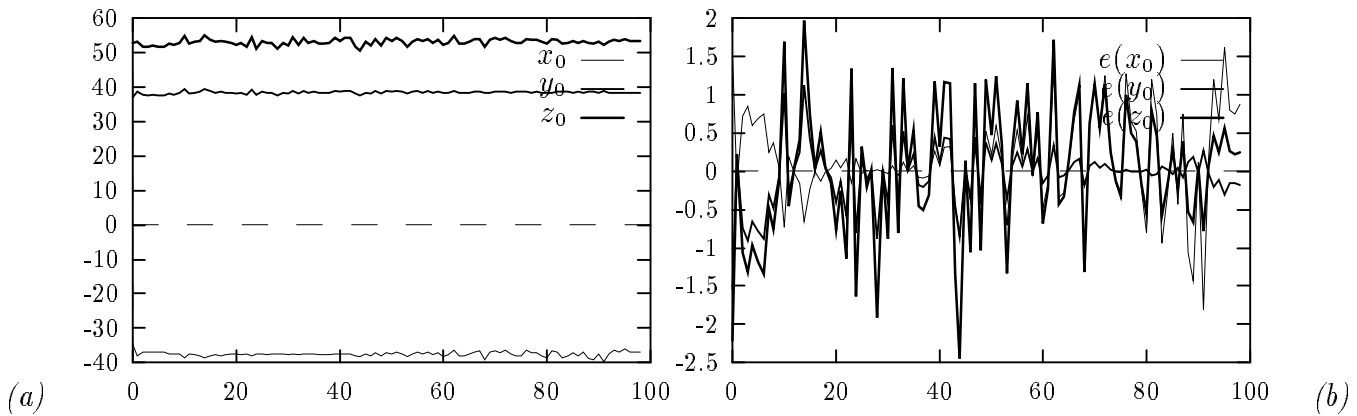


Figure 3.3 : *Reconstruction 3D d'un point en supprimant l'erreur de discrétisation : (a) valeurs estimées à chaque itération des coordonnées 3D du point. (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*
(avec bruit)

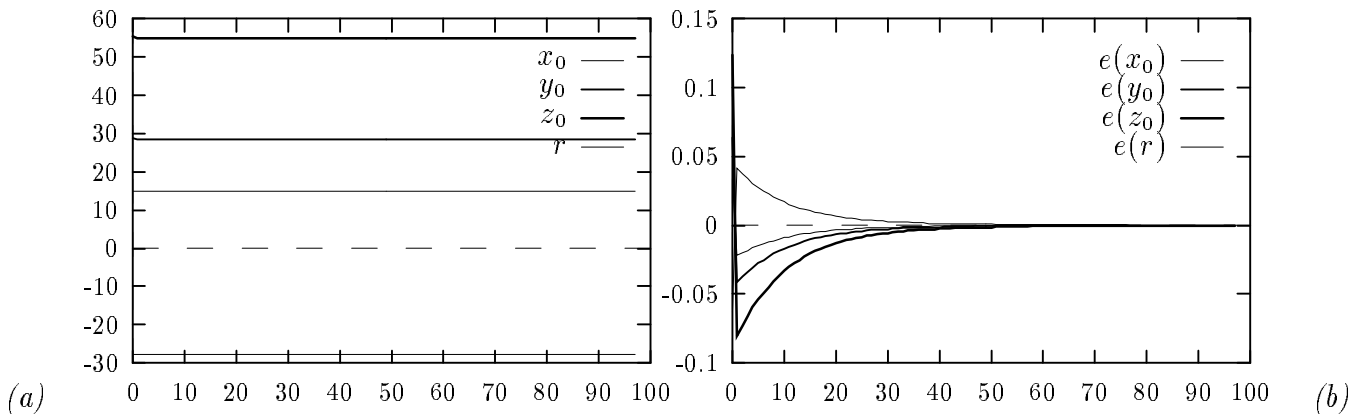


Figure 3.4 : *Reconstruction d'une sphère en supprimant l'erreur de discrétisation : (a) valeurs estimées à chaque itération des coordonnées x_0, y_0, z_0 du centre de la sphère et de son rayon r . (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*
(sans bruit)

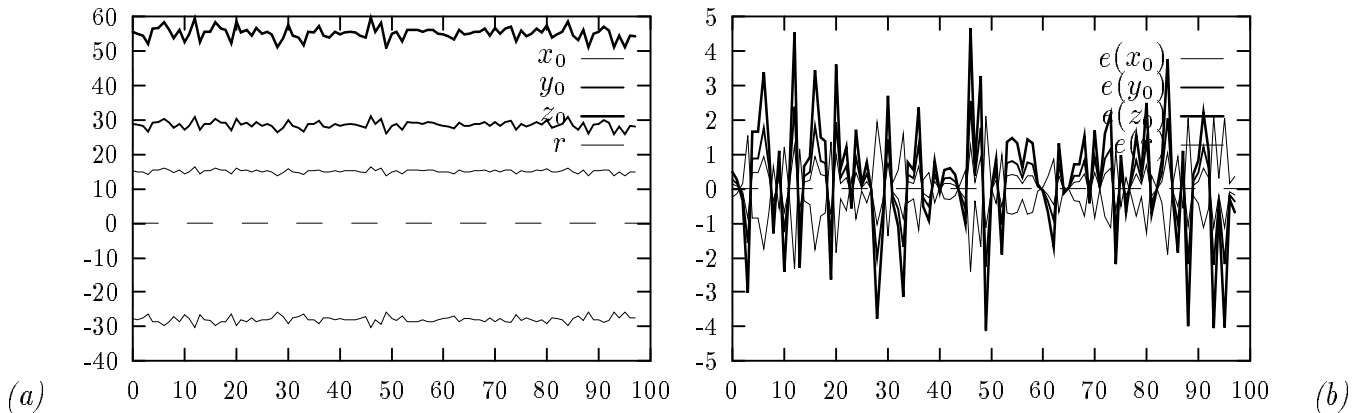


Figure 3.5 : *Reconstruction d'une sphère en supprimant l'erreur de discrétisation : (a) valeurs estimées à chaque itération des coordonnées x_0, y_0, z_0 du centre de la sphère et de son rayon r . (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*

(avec bruit)

3.4.2.3 Exemple 3 : le cylindre

Sur les figures 3.6 (sans bruit) et 3.7 (avec bruit) sont représentés les résultats correspondant à la reconstruction d'un cylindre en appliquant la même stratégie d'annulation des effets des erreurs de discrétisation.

Ces différents résultats révèlent l'intérêt d'imposer des contraintes sur le mouvement de la caméra pour estimer la structure des primitives. Signalons que des résultats similaires ont été obtenus pour les primitives de type droite et cercle. Nous allons voir à présent que l'on peut améliorer les résultats obtenus en présence de bruit en se focalisant sur l'aspect robustesse de la reconstruction.

3.4.3 Minimisation des effets des erreurs de mesure

Intuitivement, on peut penser que la configuration d'une primitive dans l'image influe sur la qualité de son estimation. Un moyen naturel d'aborder ce problème consiste à minimiser les erreurs d'estimation sur les paramètres \underline{p} , fonction des erreurs de mesure effectuées. Rappelons que \underline{p} est de la forme :

$$\underline{p} = \underline{p}(\underline{P}, \dot{\underline{P}}, T) \quad (3.10)$$

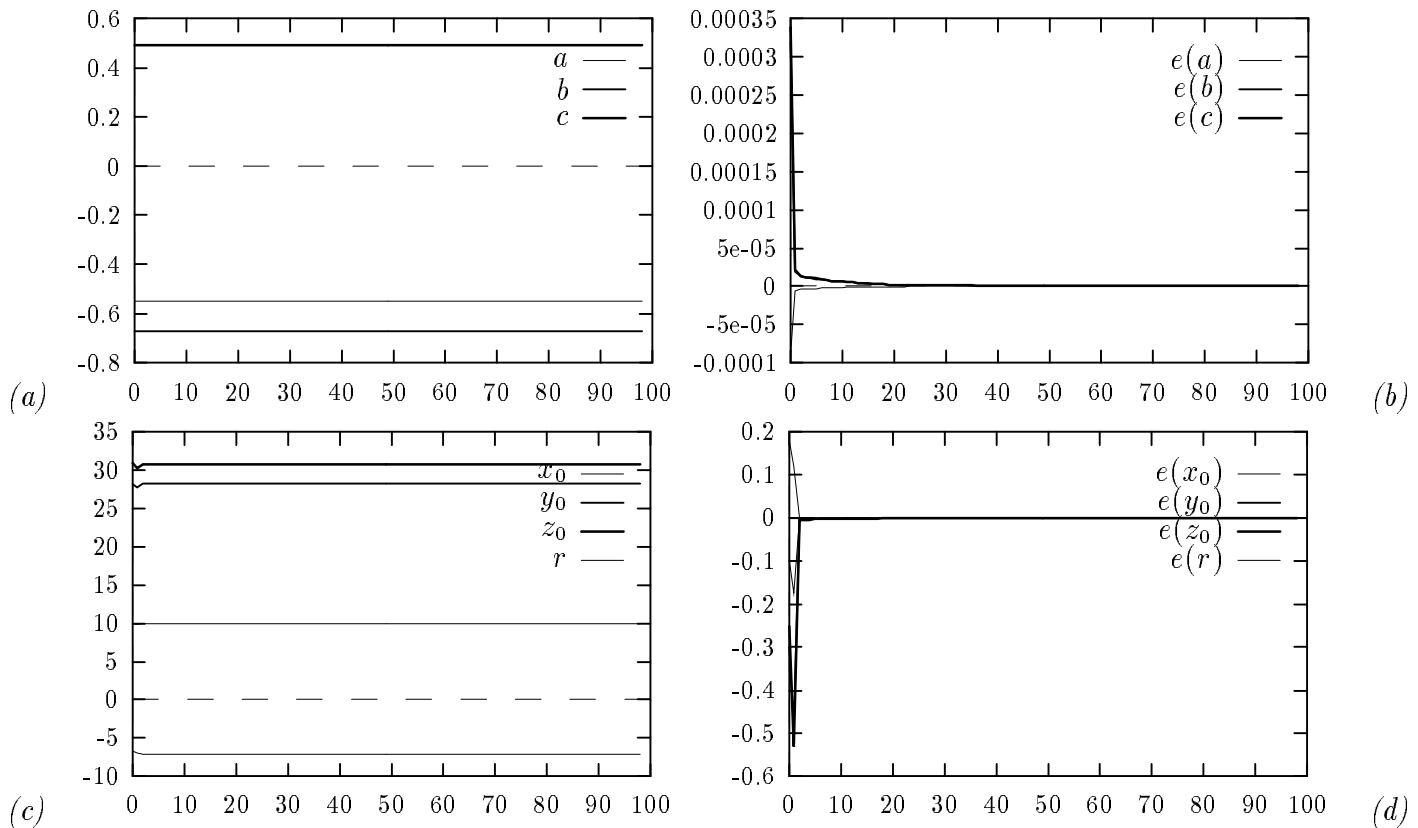


Figure 3.6 : Reconstruction d'un cylindre en supprimant l'erreur de discrétisation : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles. (c) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r . (d) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.

(sans bruit)

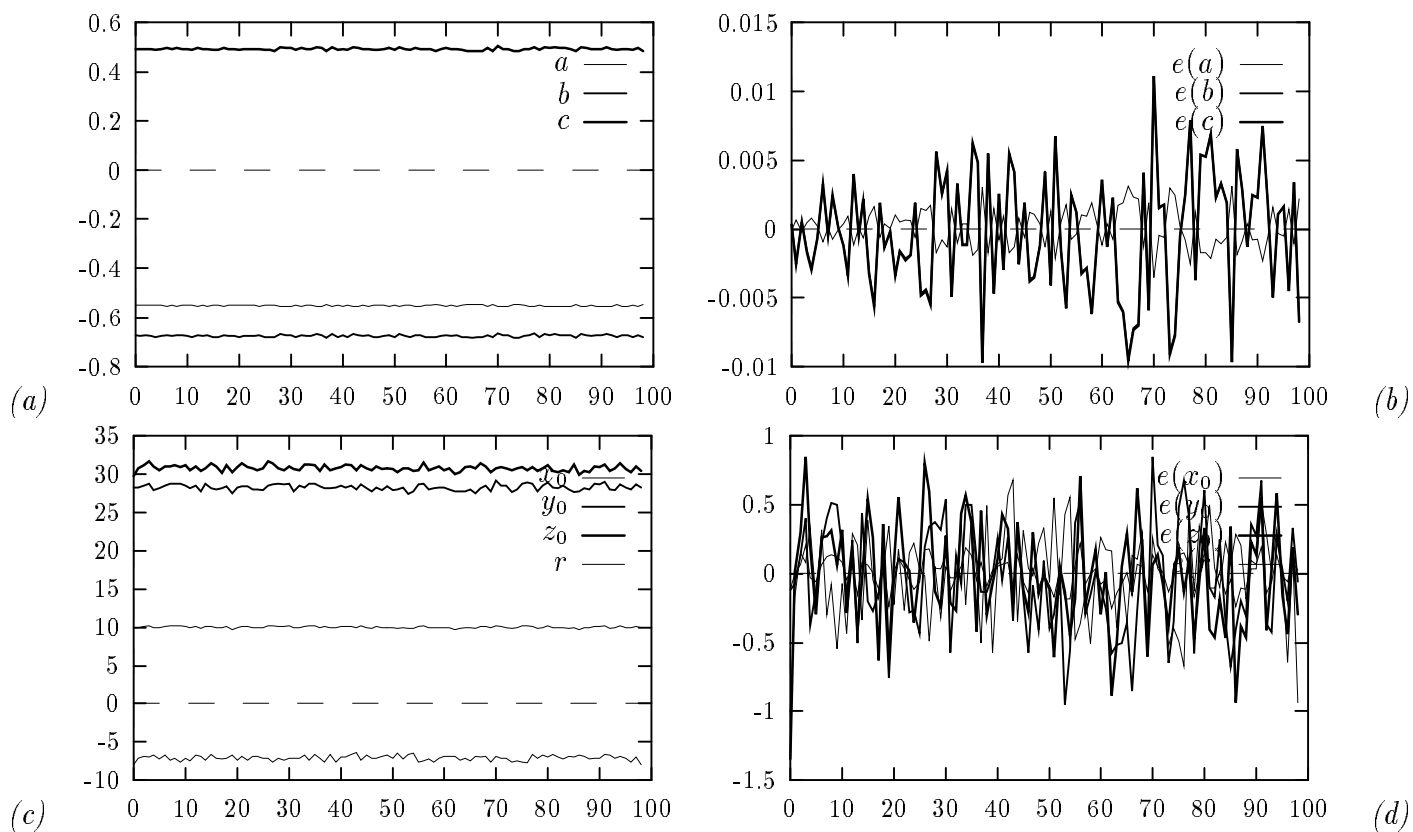


Figure 3.7 : Reconstruction d'un cylindre en supprimant l'erreur de discrétisation :
 (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles. (c) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r . (d) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.

(avec bruit)

Les erreurs sur \underline{p} sont donc étroitement liées aux erreurs de mesure sur les informations visuelles \underline{P} et $\dot{\underline{P}}$ et sur la vitesse de la caméra T . Soit p un des paramètres représentant la primitive à reconstruire. Si l'on suppose que les erreurs de mesure sur \underline{P} , $\dot{\underline{P}}$ et T sont décorréelées, l'incertitude σ_p sur l'estimation de p peut s'exprimer sous la forme suivante :

$$(\sigma_p)^2 = \sum_{i=1}^m \left(\frac{\partial p}{\partial P_i} \right)^2 (\sigma_{P_i})^2 + \sum_{j=1}^m \left(\frac{\partial p}{\partial \dot{P}_j} \right)^2 (\sigma_{\dot{P}_j})^2 + \sum_{k=1}^6 \left(\frac{\partial p}{\partial T_k} \right)^2 (\sigma_{T_k})^2 \quad (3.11)$$

Notons $p_{a_i} = \left(\frac{\partial p}{\partial a_i} \right)^2$, $a_i \in \{\underline{P}, \dot{\underline{P}}, T\}$. La minimisation de $(\sigma_p)^2$ par rapport à une configuration \underline{P} de la primitive dans l'image se ramène alors à trouver l'ensemble des paramètres P_j ($j = 1$ à m) tels que :

$$\left(\frac{\partial p_{a_i}}{\partial P_j} \right) = 0, \quad \forall a_i \text{ et } \forall j = 1 \text{ à } m \quad (3.12)$$

La résolution explicite de tels systèmes non linéaires s'avère extrêmement délicate aussi avons-nous procédé de la manière suivante :

- (1) on sélectionne les configurations de la primitive dans l'image qui semblent intuitivement entraîner une meilleure estimation (celles qui, par exemple, simplifient au mieux les expressions analytiques) ;
- (2) on vérifie ensuite si chaque équation du système (3.12) est satisfaite. On peut ainsi trouver des configurations optimales permettant une reconstruction robuste.

3.4.3.1 Cas du point

Nous cherchons à minimiser ici l'incertitude σ_z sur la profondeur z du point à reconstruire. Celle-ci peut être calculée à partir de l'équation (2.14) en utilisant la relation (3.11) avec $\underline{P} = (X, Y)$. La configuration choisie comme candidat à fournir une estimation de z robuste est bien évidemment telle que le point apparaisse centré dans l'image, soit $\underline{P} = (0, 0)$. On peut montrer que :

$$X = Y = 0 \text{ et } \dot{X} = \dot{Y} = 0 \text{ et } V_z = \Omega_z = 0 \Rightarrow \frac{\partial p_{a_i}}{\partial X} = \frac{\partial p_{a_i}}{\partial Y} = 0 \quad (3.13)$$

Outre les contraintes sur le mouvement de la caméra pour conserver en permanence la configuration $X = Y = 0$, des contraintes supplémentaires ($V_z = \Omega_z = 0$)

sont donc nécessaires pour que cette configuration minimise les effets des erreurs de mesure sur la reconstruction du point. Signalons que s'il nous a été impossible de prouver que cette configuration est le seul minimum (en raison de la complexité du système (3.12)), nous avons pu vérifier que certaines configurations (par exemple $X = 1, Y = 0$) n'annulaient pas, quels que soient les mouvements de la caméra, l'ensemble des coefficients $\frac{\partial p_{ai}}{\partial P_j}$ et ne minimisaient donc pas l'incertitude sur z .

Le fait que le point central soit un point de l'image ayant des propriétés particulières peut paraître surprenant si l'on considère une caméra sphérique au lieu d'un plan image. En effet, sur une image sphérique, tous les points ont les mêmes propriétés. Cependant, pour obtenir cette image sphérique à partir d'une seule image plane, un changement de variable non linéaire sauf pour le point central est nécessaire. Celui-ci a la forme suivante :

$$\begin{cases} X_s = \frac{X}{\sqrt{1 + X^2 + Y^2}} \\ Y_s = \frac{Y}{\sqrt{1 + X^2 + Y^2}} \end{cases} \quad (3.14)$$

où X_s et Y_s sont les coordonnées du point sur l'image sphérique.

On peut facilement vérifier que les termes $\left(\frac{\partial X}{\partial X_s}\right)^2, \left(\frac{\partial X}{\partial Y_s}\right)^2, \left(\frac{\partial Y}{\partial X_s}\right)^2, \left(\frac{\partial Y}{\partial Y_s}\right)^2$ sont nuls si, et seulement si, on a $X_s = Y_s = X = Y = 0$ (car le plan image n'est tangent à l'image sphérique que pour le point central). Ainsi, en utilisant un point d'une image sphérique, les effets des erreurs de mesure sur ce point seront minimaux si ce point est le point central, ce qui confirme bien le résultat que nous avons obtenu. On peut également rappeler que de nombreux travaux, notamment sur la tête binoculaire active [SAN86], [BAN86], [ALO87], [PAH92] ont mis en évidence l'intérêt d'utiliser le point de fixation qui n'est rien d'autre que le point central dans notre cas monoculaire.

- **Remarque** : Notons qu'en supposant que l'incertitude σ_z sur la profondeur z ne dépend que des erreurs sur les informations visuelles \underline{P} et $\dot{\underline{P}}$, les configurations dans l'image du point autres que la configuration ($X = Y = 0$) telles que par exemple ($X = 1, Y = 0$) ne minimisent toujours pas σ_z .

3.4.3.2 Autres primitives

Le même principe a été appliqué aux autres primitives que nous avons étudiées. Nous avons effectué, pour toutes ces primitives, la minimisation des incertitudes

sur les paramètres \underline{p}_0 du plan des limbes par rapport aux paramètres 2D adéquats. Cette procédure n'est pas applicable dans le cas où l'estimation des paramètres 3D du cylindre est effectuée en utilisant seulement un de ses limbes, le plan des limbes ne pouvant alors être entièrement déterminé. Par conséquent, nous avons choisi, pour ce cas particulier, de minimiser les incertitudes sur les paramètres a, b et c du vecteur directeur de l'axe du cylindre.

Nous avons obtenu les résultats suivants :

- la reconstruction d'une droite est optimale quand celle-ci apparaît centrée horizontale ou verticale dans l'image ($\rho = 0, \theta = k\frac{\pi}{2}$) avec en outre $V_z = 0$ et $V_x = \Omega_y = 0$ si la droite est horizontale ou $V_y = \Omega_x = 0$ si elle est verticale.
- la reconstruction d'une sphère est optimale quand sa projection dans l'image est un cercle centré ($X_c = Y_c = \mu_{11} = 0, \mu_{20} = \mu_{02}$) si, en outre, $\Omega_z = 0$.
- la reconstruction d'un cylindre en utilisant ses deux limbes est optimale quand celui-ci se projette en deux droites symétriques parallèles verticales ou horizontales dans l'image ($\rho_1 = -\rho_2, \theta_1 = \theta_2 = k\frac{\pi}{2}$) avec en outre $V_x = 0$ si les droites sont horizontales ou $V_y = 0$ si elles sont verticales.
- la reconstruction d'un cylindre en utilisant un seul de ses limbes est optimale quand celui-ci se projette en une droite centrée horizontale ou verticale dans l'image ($\rho_1 = 0, \theta_1 = k\frac{\pi}{2}$) avec en outre $\Omega_y = 0$ si la droite est horizontale et $\Omega_x = 0$ si elle est verticale.
- **Remarque** : Nous avons également vérifié que, comme dans le cas de la sphère, la reconstruction d'un cercle est optimale quand sa projection dans l'image est un cercle centré. Cependant, cette position n'est jamais atteignable par asservissement visuel. En effet, en raison d'une singularité isolée dans la matrice d'interaction associée au cercle lorsque l'image du cercle est un cercle centré, il n'est pas possible d'établir de loi de commande permettant de faire converger la caméra vers une position telle que le cercle se projette en un cercle centré dans l'image [CHA90] [ESP92]. Les mouvements secondaires nécessaires à la reconstruction du cercle sont de plus impossibles à réaliser (ils ne conservent pas l'image du cercle sous la forme d'un cercle centré), ceci, toujours en raison de la singularité isolée.

L'approche asservissement visuel est ici encore parfaitement adaptée pour générer automatiquement les mouvements de la caméra permettant d'obtenir des configurations particulières entre la caméra et la primitive considérée (tels par exemple

que l'image d'un point soit au centre de l'image ou l'image d'une sphère soit un cercle centré).

3.4.4 Résultats de simulation

3.4.4.1 Exemple 1 : le point

La position optimale pour la reconstruction d'un point est telle qu'il apparaisse centré dans l'image comme nous l'avons montré au paragraphe 3.4.3.1 (voir figure 3.8). A partir de la position initiale de la figure 2.8, une loi de commande détermine les mouvements de la caméra permettant de converger vers la position désirée. La figure 3.9 montre les résultats de la reconstruction du point obtenus pendant et après la phase de convergence (celle-ci est assurée aux alentours de l'itération 15).

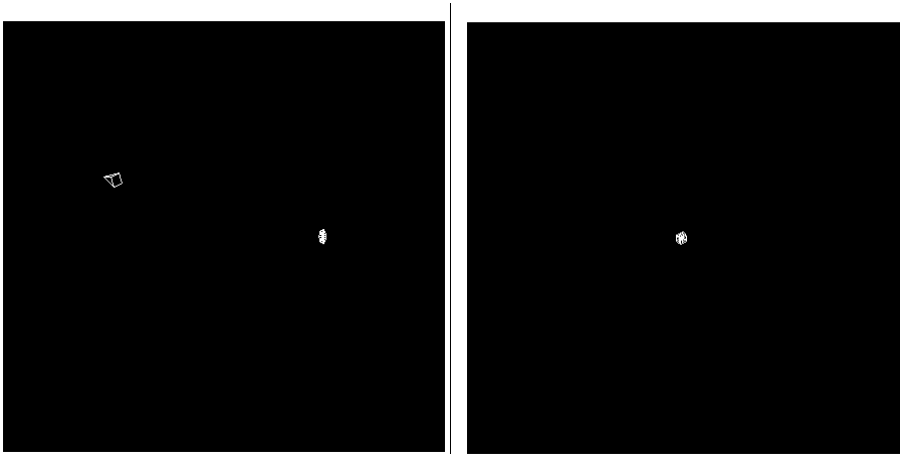


Figure 3.8 : *Position optimale entre la caméra et le point*

On peut observer que ces résultats sont assez médiocres pendant la phase de convergence. Ceci s'explique par le fait que, pendant cette phase, les mouvements de la caméra ne respectent pas les contraintes nécessaires pour annuler les effets de discrétisation. La contrainte $\dot{\underline{P}} = 0$ ne peut en effet être respectée puisque l'on souhaite amener \underline{P} à une valeur désirée différente \underline{P}_d . Par contre, lorsque la convergence est assurée, les mouvements de la caméra sont contraints pour conserver $\underline{P} = \underline{P}_d$ (d'où $\dot{\underline{P}} = 0$). De plus, il est également possible d'assurer $\dot{\underline{p}}_0 = 0$ ainsi que les contraintes supplémentaires établies précédemment pour minimiser les effets des erreurs de mesure (par exemple $\Omega_z = 0$ dans le cas de la sphère). On peut vérifier que les résultats obtenus après la phase de convergence sont nettement plus robustes

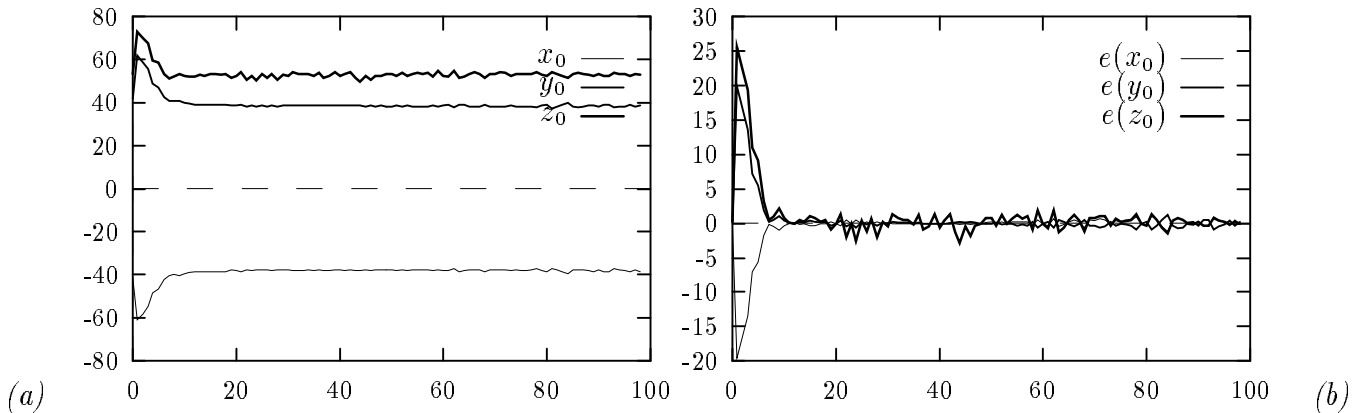


Figure 3.9 : *Reconstruction d'un point en minimisant les effets des erreurs de mesure : (a) valeurs estimées à chaque itération des coordonnées 3D du point. (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*
(avec bruit)

que ceux obtenus en appliquant seulement la stratégie précédente où l'on supprimait l'erreur de discrétisation pour une configuration quelconque entre la caméra et la primitive à reconstruire (voir figure 3.3).

3.4.4.2 Exemple 2 : la sphère

Le même principe est appliqué dans le cas de la sphère pour amener la caméra de la position visualisée figure 2.10 à celle visualisée figure 3.10 telle que l'image de la sphère soit un cercle centré. Les résultats de la reconstruction sont représentés sur la figure 3.11.

Ces courbes ont la même allure que celles correspondant à la reconstruction d'un point. Des erreurs importantes sont observées avant la convergence vers la configuration optimale de la sphère dans l'image puis, après convergence, on observe une très nette amélioration. Ici aussi, on peut remarquer que l'estimation des paramètres de la sphère est plus robuste que dans le cas où l'on ne supprimait que l'erreur de discrétisation (voir figure 3.5).

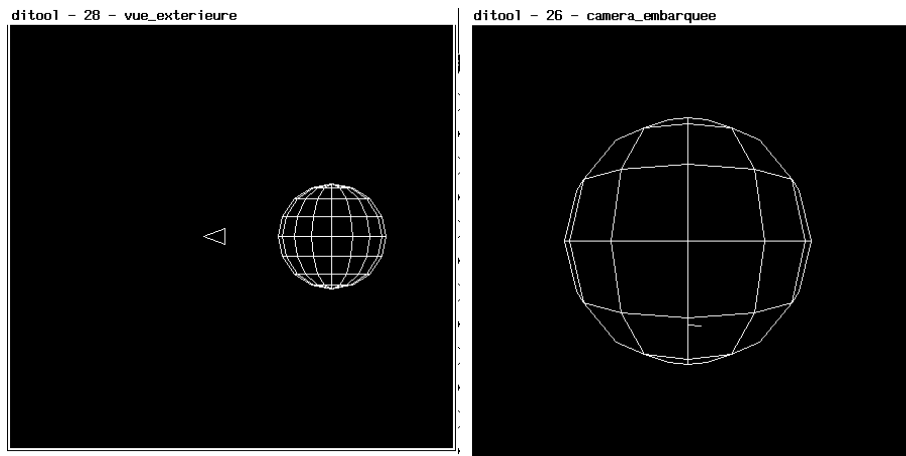


Figure 3.10 : *Position optimale entre la caméra et la sphère*

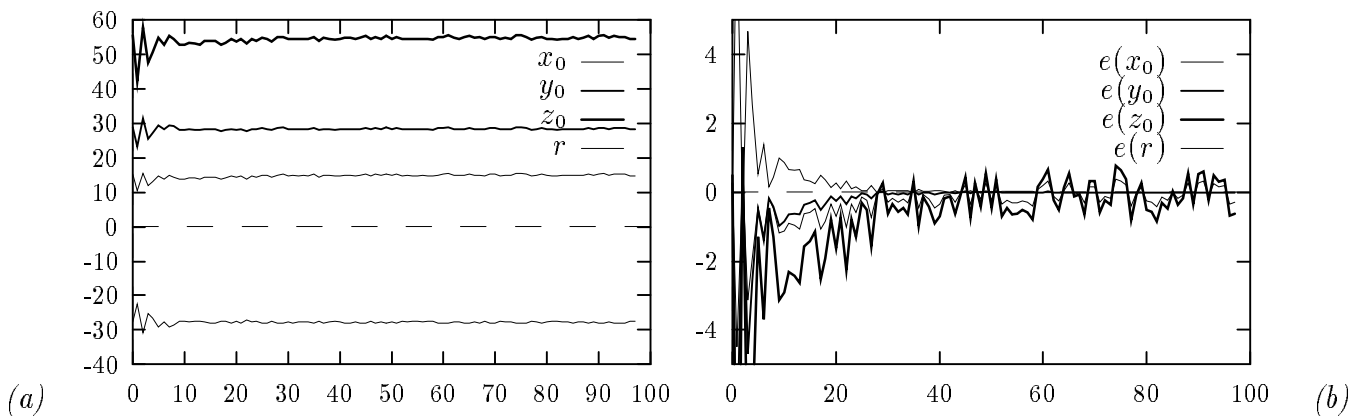


Figure 3.11 : *Reconstruction d'une sphère en minimisant les effets des erreurs de mesure : (a) valeurs estimées à chaque itération des coordonnées x_0, y_0, z_0 du centre de la sphère et de son rayon r . (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*

(avec bruit)

3.4.4.3 Exemple 3 : le cylindre

Nous avons choisi comme position optimale pour la reconstruction celle qui donne pour image du cylindre deux droites symétriques verticales (voir figure 3.12). La figure 3.13 montre les résultats de la reconstruction du cylindre obtenus en utilisant cette stratégie.

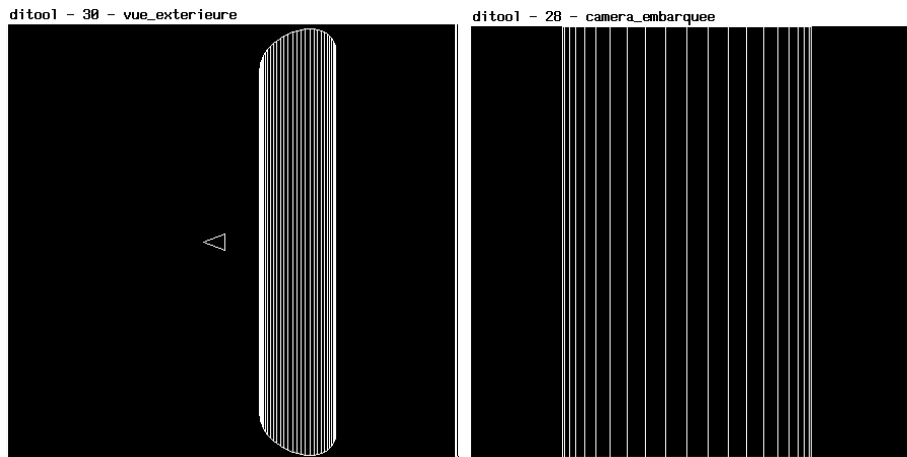


Figure 3.12 : *Position optimale entre la caméra et le cylindre*

Le même phénomène que dans les deux cas précédents est observé sur ces courbes. On notera une fois de plus l'amélioration apportée par cette stratégie par rapport à la stratégie précédente (voir figure 3.7).

- **Remarques :**

- (1) Des résultats similaires ont été obtenus pour les primitives de type droite. Par contre, comme nous l'avons déjà vu, cette stratégie n'est pas applicable au cas du cercle.
- (2) Soulignons enfin qu'il ne nous a pas semblé nécessaire de faire appel à un schéma d'estimation linéaire optimal tel que le filtre de Kalman. Un tel filtrage aurait en effet ralenti la convergence de notre méthode d'estimation vers les valeurs réelles des paramètres 3D puisque la loi de commande, en convergeant vers une configuration optimale, permet d'améliorer les estimées des paramètres au cours du temps. Un simple filtre moyenneur, par exemple sur les trois ou cinq dernières valeurs, semble plus adéquat pour robustifier encore davantage les résultats obtenus.

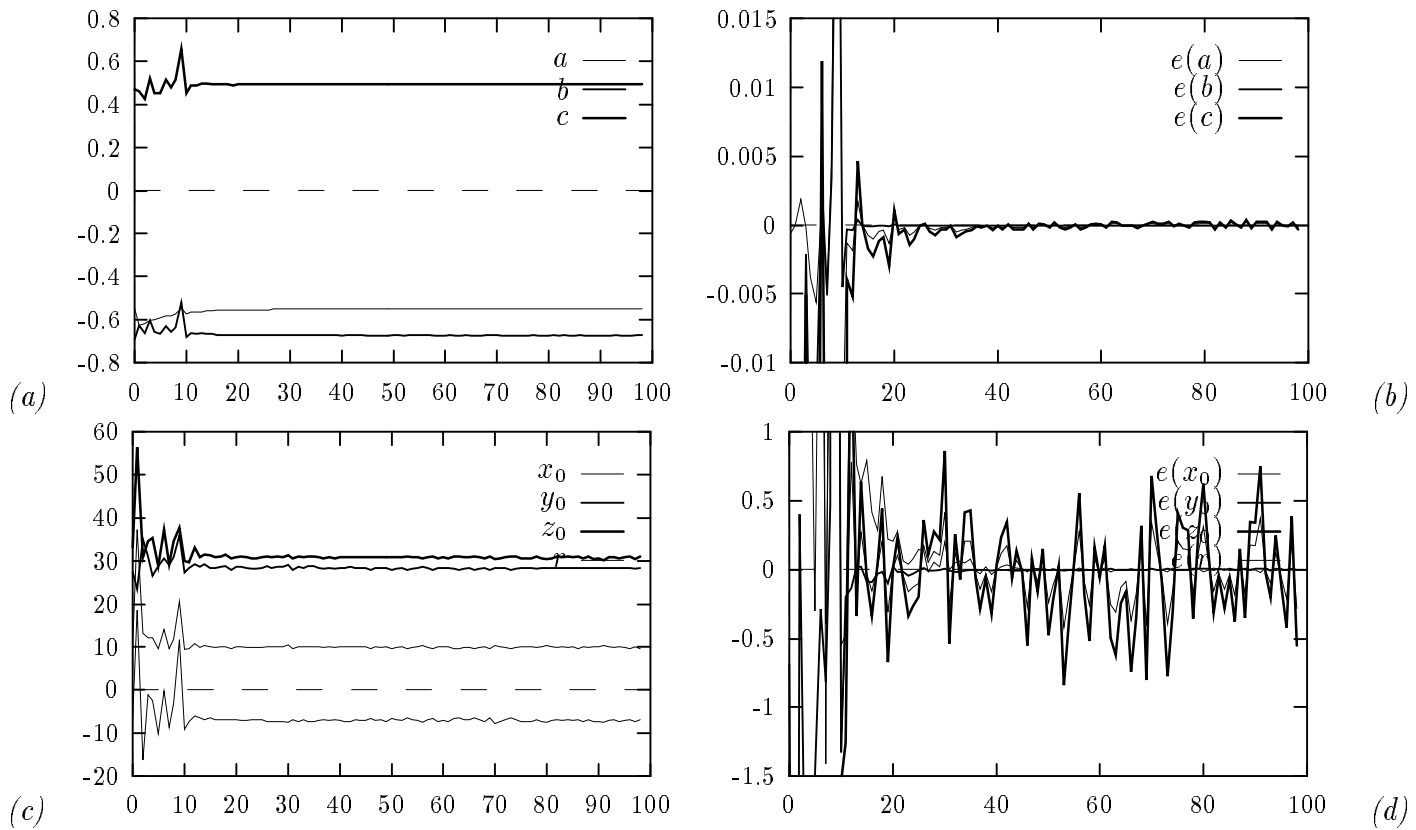


Figure 3.13 : *Reconstruction d'un cylindre en minimisant les effets des erreurs de mesure : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles. (c) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r . (d) erreurs entre les valeurs estimées de ces paramètres et leur valeurs réelles.*

(avec bruit)

3.4.5 La commande référencée vision

Nous venons de montrer qu'on pouvait améliorer la connaissance d'une scène par des mouvements adéquats de la caméra. Une commande en boucle fermée sur des informations visuelles telle la commande référencée vision [WEI87] [FED89] [RIV89] [ESP92] [CHA90] paraît donc parfaitement adaptée pour générer automatiquement ces différents mouvements. Nous n'en rappelons ici que les principes de base et renvoyons le lecteur à [SAM91] pour une étude complète de la commande des robots manipulateurs par l'approche fonction de tâche, à [SAM90] pour l'application de cette approche à la commande référencée capteurs et à [ESP92], [RIV89] ou [CHA90] pour son application à la commande référencée vision.

Nous avons défini dans ce qui précède des contraintes sur le mouvement de la caméra afin d'obtenir une meilleure reconstruction des primitives géométriques.

Plus précisément, pour supprimer les erreurs de discrétisation, les mouvements de la caméra doivent être tels que $\dot{\underline{P}} = 0, \forall t$, ce qui peut s'exprimer sous la forme $\underline{P}(t) - \underline{P}_d = 0, \forall t$ où $\underline{P}(t)$ est la valeur mesurée à chaque itération des m paramètres représentant la primitive dans l'image et où \underline{P}_d est la valeur de \underline{P} mesurée à la première itération (on aurait également pu choisir pour \underline{P}_d la valeur de \underline{P} mesurée à l'itération précédente). De même, pour minimiser les effets des erreurs de mesure, les mouvements de la caméra doivent assurer $\underline{P}(t) - \underline{P}_d = 0, \forall t$ où \underline{P}_d est ici la configuration optimale choisie de la primitive dans l'image (par exemple, pour le cas du point, $X = Y = 0$).

Dans les deux cas, la fonction de tâche correspondante s'exprime sous la forme [SAM91] [ESP92] [CHA90] :

$$\underline{e} = W^+ C (\underline{P}(t) - \underline{P}_d) + (\mathbb{I}_6 - W^+ W) \underline{e}_2 \quad (3.15)$$

où :

- \underline{e} est la fonction de tâche que l'on souhaite réguler à zéro (e est de dimension 6 puisque l'on considère le cas général où la caméra a six degrés de liberté) ;
- C est une matrice de dimension $m \times m$ qui, idéalement, s'écrit :

$$C = W L_{\underline{P}}^{T+} (\underline{P}, \underline{p}_0) \quad (3.16)$$

avec :

- $L_{\underline{P}}^{T+}$, de dimension $6 \times m$, la pseudo-inverse de la matrice d'interaction associée à \underline{P} (qui correspond grossièrement au jacobien inverse de la tâche référencée vision à accomplir) ;

- W , de dimension $m \times 6$, une matrice de rang plein telle que $\text{Ker } W = \text{Ker } L_{\underline{P}}^T$.

Remarque : Les matrices $L_{\underline{P}}^{T+}$ et W dépendent des paramètres \underline{P} qui sont mesurés dans l'image mais également des paramètres \underline{p}_0 , a priori inconnus, représentant la surface des limbes de la primitive. La méthode de reconstruction que nous avons détaillée au chapitre précédent permet d'en avoir une estimation $\hat{\underline{p}}_0$ qui est bien entendu utilisée pour calculer à chaque itération $L_{\underline{P}}^{T+}$ et W .

- \underline{e}_2 est une tâche secondaire (dans notre cas un suivi de trajectoire) construite afin de permettre des mouvements de la caméra tout en assurant la contrainte $\underline{P}(t) = \underline{P}_d$. Des déplacements de la caméra sont en effet en permanence indispensables pour pouvoir estimer la structure des primitives. Par exemple, pour le cas de la sphère (qui doit apparaître sous la forme d'un cercle centré dans l'image), la tâche secondaire va consister à déplacer la caméra sur une sphère concentrique afin de reconstruire la sphère tout en l'observant sous la forme d'un cercle centré dans l'image. Cette tâche secondaire permet également d'assurer les contraintes supplémentaires que nous avons établies, à savoir $\underline{p}_0 = 0$ pour supprimer les erreurs de discrétisation et celles du type $V_z = \Omega_z = 0$ (pour le cas du point) qui fournissent une estimation plus robuste.
- W^+ et $\mathbb{I}_6 - W^+W$ sont des opérateurs qui assurent la compatibilité de la tâche secondaire \underline{e}_2 avec la contrainte $\underline{P}(t) = \underline{P}_d$

La fonction de tâche \underline{e} étant construite, une commande simplifiée, qui calcule la vitesse désirée de la caméra T_c et assure une décroissance exponentielle de \underline{e} , est donnée par [ESP92] :

$$T_c = -\lambda \underline{e} - \frac{\partial \underline{e}}{\partial t}, \quad \lambda > 0 \quad (3.17)$$

où λ est un gain fixé par l'utilisateur dépendant de la fréquence d'échantillonnage de la boucle de commande et où $\frac{\partial \underline{e}}{\partial t}$ peut ici être approximée par $(\mathbb{I}_6 - W^+W) \frac{\partial \underline{e}_0}{\partial t}$ puisque la scène est supposée fixe (on suppose en outre que W et C sont lentement variables de telle sorte que leur variation soit négligeable).

Nous détaillons à présent la construction de la fonction de tâche correspondant à la reconstruction du point.

3.4.5.1 Construction de la fonction de tâche correspondant à la reconstruction d'un point

Dans le cas du point, on a $\underline{P} = (X, Y)$ et la matrice d'interaction associée s'exprime par (voir (2.12)) :

$$L_{\underline{P}}^T = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \quad (3.18)$$

$L_{\underline{P}}^T$ est toujours de rang plein 2 aussi choisit-on $W = L_{\underline{P}}^T$. On peut par ailleurs facilement calculer la pseudo-inverse $L_{\underline{P}}^{T+}$ de $L_{\underline{P}}^T$ (vérifiant $L_{\underline{P}}^T L_{\underline{P}}^{T+} = \mathbb{I}_2$) et on obtient :

$$L_{\underline{P}}^{T+} = W^+ = \frac{1}{D} \begin{pmatrix} -z(1+Y^2) & zXY \\ zXY & -z(1+X^2) \\ zX & zY \\ 0 & z^2E \\ -z^2E & 0 \\ z^2YE & -z^2XE \end{pmatrix} \quad (3.19)$$

avec $D = E(1+z^2E)$ et $E = (1+X^2+Y^2)$. La tâche secondaire $\underline{\epsilon}_2$ est ici choisie égale à :

$$\underline{\epsilon}_2 = \begin{pmatrix} x - x_0 - V_1 t \\ y - y_0 - V_2 t \\ z - z_0 + (XV_1 + YV_2)t \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.20)$$

où x_0, y_0 et z_0 sont les positions initiales de la caméra et où V_1 et V_2 sont deux constantes choisies par l'utilisateur afin d'effectuer un mouvement à vitesse constante dans les directions \vec{x} et \vec{y} du repère de la caméra (l'un de ces deux mouvements au moins est nécessaire pour permettre la reconstruction du point). Le choix de la troisième composante de $\underline{\epsilon}_2$ permet, lui, d'assurer la contrainte supplémentaire $\dot{z} = 0$ supprimant les erreurs de discrétisation. En effet, si l'on suppose pour simplifier les calculs que la tâche est parfaitement réalisée ($\underline{\epsilon} = 0$), la consigne de la caméra prend pour valeur :

$$T_c = \begin{pmatrix} V_x \\ V_y \\ V_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} = -(\mathbb{I}_6 - W^+W) \frac{\partial \underline{\epsilon}_2}{\partial t} = (\mathbb{I}_6 - W^+W) \begin{pmatrix} V_1 \\ V_2 \\ -XV_1 - YV_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.21)$$

avec :

$$\mathbb{I}_6 - W^+W = \frac{1}{D} \begin{pmatrix} D-1-Y^2 & XY & X & 0 & -zE & zYE \\ XY & D-1-X^2 & Y & zE & 0 & -zXE \\ X & Y & D-X^2-Y^2 & -zYE & zXE & 0 \\ 0 & zE & -zYE & (1+z^2X^2)E & z^2XYE & z^2XE \\ -zE & 0 & zXE & z^2XYE & (1+z^2Y^2)E & z^2YE \\ zYE & -zXE & 0 & z^2XE & z^2YE & (1+z^2)E \end{pmatrix}$$

et on peut alors vérifier que $\dot{z} = -V_z - zY\Omega_x + zX\Omega_y = 0$.

Pour le cas particulier du point centré dans l'image ($X_d = Y_d = 0$), on a lorsque la tâche est parfaitement réalisée ($X = Y = 0$) :

$$T_c = \frac{1}{1+z^2} \begin{pmatrix} -z^2 & 0 & 0 & 0 & -z & 0 \\ 0 & -z^2 & 0 & z & 0 & 0 \\ 0 & 0 & 1+z^2 & 0 & 0 & 0 \\ 0 & z & 0 & 1 & 0 & 0 \\ -z & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+z^2 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{z}{1+z^2} \begin{pmatrix} -zV_1 \\ -zV_2 \\ 0 \\ V_2 \\ -V_1 \\ 0 \end{pmatrix} \quad (3.22)$$

qui permet d'animer la caméra d'un mouvement à vitesse constante dans la direction \vec{x} d'amplitude $-z^2V_1/(1+z^2)$ (respectivement $-z^2V_2/(1+z^2)$ dans la direction \vec{y}) compensé par une rotation autour de Ω_y (respectivement Ω_x) pour conserver l'image du point au centre de l'image. La contrainte $\dot{z} = 0$ est bien entendu assurée, de même que $V_z = \Omega_z = 0$ qui minimisent les effets des erreurs de mesure.

3.4.5.2 Construction de la fonction de tâche correspondant à la reconstruction d'une droite

Dans le cas de la droite, on a $\underline{P} = (\rho, \theta)$ et la matrice d'interaction associée $L_{\underline{P}}^T$ est de rang plein 2 (voir équation (2.22)). On choisit donc également $W = L_{\underline{P}}^T$.

Comme pour le point, la condition $\dot{\underline{P}} = 0$ n'entraîne pas nécessairement $\underline{\dot{p}}_0 = 0$. Nous recherchons donc ici la consigne T_c de la caméra qui va assurer la contrainte $\underline{\dot{p}}_0 = 0$.

Déterminons tout d'abord la matrice d'interaction $L_{\underline{p}_0}^T$ associée à \underline{p}_0 . Sachant que $\underline{p}_0 = (a, b, c) = (-a_2/d_2, -b_2/d_2, -c_2/d_2)$ (voir paragraphe 2.3.2) on a :

$$\underline{\dot{p}}_0 = \frac{1}{d_2^2} \begin{pmatrix} a_2\dot{d}_2 - \dot{a}_2d_2 \\ b_2\dot{d}_2 - \dot{b}_2d_2 \\ c_2\dot{d}_2 - \dot{c}_2d_2 \end{pmatrix} \quad (3.23)$$

où $\dot{a}_2, \dot{b}_2, \dot{c}_2, \dot{d}_2$ sont déterminées de la manière suivante. On différentie par rapport au temps t les équations (2.15) des deux plans définissant la droite ainsi que les trois contraintes (2.16). On choisit alors deux points de la droite 3D, par exemple : $m_1 = (x_1, y_1, z_1)^t = (0, \frac{c_1 d_2}{b_1 c_2 - c_1 b_2}, \frac{-b_1 d_2}{b_1 c_2 - c_1 b_2})^t$ et $m_2 = (x_2, y_2, z_2)^t = (\frac{b_1 d_2}{a_1 b_2 - a_2 b_1}, \frac{-a_1 d_2}{a_1 b_2 - a_2 b_1}, 0)^t$.

Sachant que la vitesse $V(m)$ d'un point m de coordonnées $\underline{x} = (x, y, z)^t$ exprimée dans le repère caméra (voir figure 2.1) s'exprime par :

$$V(m) = -V(O) - \Omega \wedge \underline{x} \quad (3.24)$$

où $T = (V(O), \Omega)$ est le torseur cinématique de la caméra, on a :

$$\begin{cases} \dot{x}_1 = -V_x - z_1 \Omega_y + y_1 \Omega_z & ; & \dot{x}_2 = -V_x + y_2 \Omega_z \\ \dot{y}_1 = -V_y + z_1 \Omega_x & ; & \dot{y}_2 = -V_y - x_2 \Omega_z \\ \dot{z}_1 = -V_z - y_1 \Omega_x & ; & \dot{z}_2 = -V_z - y_2 \Omega_x + x_2 \Omega_y \end{cases} \quad (3.25)$$

On peut alors facilement déterminer, par la résolution d'un système linéaire, les valeurs de $\dot{a}_2, \dot{b}_2, \dot{c}_2, \dot{d}_2$. En substituant ces valeurs dans (3.23), on obtient :

$$L_{\underline{p}_0}^T = \frac{1}{d_2^6} \begin{pmatrix} a_2^2 - a_1^2 & a_2 b_2 - a_1 b_1 & a_2 c_2 - a_1 c_1 & 0 & c_2 d_2 & -b_2 d_2 \\ a_2 b_2 - a_1 b_1 & b_2^2 - b_1^2 & b_2 c_2 - b_1 c_1 & -c_2 d_2 & 0 & a_2 d_2 \\ a_2 c_2 - a_1 c_1 & b_2 c_2 - b_1 c_1 & c_2^2 - c_1^2 & b_2 d_2 & -a_2 d_2 & 0 \end{pmatrix} \quad (3.26)$$

Pour que la condition $\underline{\dot{p}}_0 = 0$ soit satisfaite on doit avoir :

$$L_{\underline{p}_0}^T T_c = L_{\underline{p}_0}^T (\mathbb{I}_6 - W^+ W) \frac{\partial \underline{\epsilon}_2}{\partial t} = L_{\underline{p}_0}^T (\mathbb{I}_6 - W^+ W) \begin{pmatrix} V1 \\ V2 \\ V3 \\ V4 \\ V5 \\ V6 \end{pmatrix} = 0 \quad (3.27)$$

soit :

$$\frac{1}{d_2^6} \begin{pmatrix} a_2^2 & a_2 b_2 & a_2 c_2 & a_1 d_2 H_1 & b_1 d_2 H_1 & c_1 d_2 H_1 \\ a_2 b_2 & b_2^2 & b_2 c_2 & -a_1 d_2 H_2 & -b_1 d_2 H_2 & -c_1 d_2 H_2 \\ a_2 c_2 & b_2 c_2 & c_2^2 & a_1 d_2 H_3 & b_1 d_2 H_3 & c_1 d_2 H_3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \\ V4 \\ V5 \\ V6 \end{pmatrix} = 0 \quad (3.28)$$

où $H_1 = b_1c_2 - b_2c_1$, $H_2 = a_1c_2 - a_2c_1$ et $H_3 = a_1b_2 - a_2b_1$. Ce qui donne :

$$\frac{\partial \underline{c}_2}{\partial t} = \begin{pmatrix} -\frac{a_2V_1+b_2V_2}{c_2} \\ V_5 \\ V_6 \\ V_2 \\ V_1 \\ -\frac{V_6c_1+V_5b_1}{a_1} \end{pmatrix} \quad (3.29)$$

On peut alors en déduire la consigne T_c de la caméra. Pour le cas particulier où la droite apparaît verticale centrée dans l'image ($b_1 = c_1 = a_2 = 0$), on a lorsque la tâche est parfaitement réalisée :

$$T_c = \frac{-1}{b_2(d_2^2 + 1)} \begin{pmatrix} -b_2d_2(b_2V_6 - c_2V_5 - d_2V_1) \\ -c_2(d_2^2 + 1)V_3 \\ b_2(d_2^2 + 1)V_3 \\ 0 \\ b_2c_2(-b_2V_6 + c_2V_5 + d_2V_1) \\ -(c_2^2 - 1)(b_2V_6 - c_2V_5 - d_2V_1) \end{pmatrix} \quad (3.30)$$

Pour assurer les contraintes $V_z = V_y = 0$ il suffit alors de poser $V_3 = 0$.

3.4.5.3 Construction de la fonction de tâche correspondant à la reconstruction d'un cylindre en utilisant un seul limbe

Dans le cas du cylindre, on a $\underline{P} = (\rho, \theta)$ puisqu'un seul limbe est considéré et la matrice d'interaction associée peut s'écrire en utilisant (2.58), (2.59), (2.52) et (2.53) et en remplaçant $\underline{p}_0 = (n_1, n_2, n_3)$ par $(x_0/A, y_0/A, z_0/A)$ (voir paragraphe 2.3.5):

$$L_{\underline{P}}^T = \frac{1}{A} \begin{pmatrix} -I_1 \cos \theta & -I_1 \sin \theta & I_1 \rho & (1 + \rho^2)A \sin \theta & -(1 + \rho^2)A \cos \theta & 0 \\ I_2 \cos \theta & I_2 \sin \theta & -I_2 \rho & -A \rho \cos \theta & -A \rho \sin \theta & -A \end{pmatrix} \quad (3.31)$$

avec $I_1 = (\rho(x_0 \cos \theta + y_0 \sin \theta) + z_0)$, $I_2 = (y_0 \cos \theta - x_0 \sin \theta)$ et $A = (x_0^2 + y_0^2 + z_0^2 - r^2)$.

On choisit comme dans les deux cas précédents la matrice $W = L_{\underline{P}}^T$.

Contrairement au cas où deux limbes du cylindre sont observables, la condition $\underline{\dot{P}} = 0$ n'entraîne pas $\underline{\dot{p}}_0 = 0$ lorsqu'un seul limbe est pris en compte. Nous allons donc adopter exactement la même démarche que dans le cas de la droite pour assurer

la contrainte $\underline{\dot{p}}_0 = 0$. La première étape consiste à déterminer la matrice $L_{\underline{p}_0}^T$ associée à \underline{p}_0 . On a :

$$\underline{\dot{p}}_0 = \frac{1}{A^2} \begin{pmatrix} A\dot{x}_0 - x_0\dot{A} \\ A\dot{y}_0 - y_0\dot{A} \\ A\dot{z}_0 - z_0\dot{A} \end{pmatrix} \quad (3.32)$$

Comme dans le cas de la droite, $\dot{x}_0, \dot{y}_0, \dot{z}_0$ sont calculés en différentiant par rapport au temps l'équation paramétrique du cylindre (2.51) ainsi que les deux contraintes (2.52) et (2.53). Le choix de quatre points particuliers appartenant au cylindre s'impose pour que le calcul de $\dot{x}_0, \dot{y}_0, \dot{z}_0$ soit possible. Nous avons choisi les points $m_{1,2} = (x_0, y_0, z_0 \pm r\sqrt{1-c^2})^t$ et $m_{3,4} = (x_0, y_0 \pm r\sqrt{1-b^2}, z_0)^t$.

La matrice $L_{\underline{p}_0}^T$ prend alors la forme suivante :

$$L_{\underline{p}_0}^T = \frac{1}{A^2} \begin{pmatrix} A(a^2 - 1) + 2x_0^2 & Aab + 2x_0y_0 & Aac + 2x_0z_0 & 0 & -Az_0 & Ay_0 \\ Aab + 2x_0y_0 & A(b^2 - 1) + 2y_0^2 & Abc + 2y_0z_0 & Az_0 & 0 & -Ax_0 \\ Aac + 2x_0z_0 & Abc + 2y_0z_0 & A(c^2 - 1) + 2z_0^2 & -Ay_0 & Ax_0 & 0 \end{pmatrix} \quad (3.33)$$

Pour que la condition $\underline{\dot{p}}_0 = 0$ soit satisfaite la consigne de la caméra prend la forme suivante dans le cas particulier où le limbe apparaît vertical centré dans l'image ($e = 0$) :

$$T_c = \frac{1}{AJ} \begin{pmatrix} A((abA + 2x_0y_0)V_2 + (acA + 2x_0z_0)V_3) \\ AV_2 \\ AV_3 \\ (K_1V_2 + K_2V_3)/y_0 \\ -z_0(V_2(abA + 2x_0y_0) + V_3(acA + 2x_0z_0)) \\ y_0(V_2(abA + 2x_0y_0) + V_3(acA + 2x_0z_0)) \end{pmatrix} \quad (3.34)$$

où $J = (-a^2A - r^2 - x_0^2)$, $K_1 = (bcA(-2r^2 - A - y_0^2) + y_0z_0(-2r^2 + A(b^2 - 2)))$ et $K_2 = (z_0^2(-2r^2 + A(2b^2 - 2c^2 - 1)) + A((2x_0^2 + y_0^2)(1 - c^2) - 3bcy_0z_0 - Ab^2))$.

De plus, les valeurs des coefficients V_2 et V_3 doivent être choisies de manière à assurer la contrainte $\Omega_x = 0$. On doit donc avoir $V_3 = \frac{-K_1V_2}{K_2}$.

- **Remarque :** Notons que la détermination des mouvements optimaux correspondant aux autres primitives que nous avons étudiées à savoir, la sphère et le cylindre, est beaucoup plus simple que dans les cas précédents. En effet, la seule contrainte à respecter est $\underline{\dot{P}} = 0$ puisque celle-ci entraîne nécessairement $\underline{\dot{p}}_0 = 0$ pour ces primitives tridimensionnelles. On obtient :

- dans le cas du cylindre, en utilisant la méthode basée sur les deux limbes, le mouvement optimal qui assure, après convergence, les contraintes $\underline{\dot{P}}$ et $\underline{\dot{p}_0}$ est tel que $V_x = -z_0\Omega_y$ si le motif final désiré consiste en deux droites symétriques verticales et, si les deux droites sont horizontales, tel que $V_y = z_0\Omega_x$, les autres composantes étant nulles.
- Dans le cas de la sphère, la consigne adéquate, lorsque la sphère se projette en un cercle centré dans l'image, est naturellement semblable à celle du point.

3.5 Conclusion

Dans ce chapitre, nous avons tenté d'améliorer la robustesse de notre approche de reconstruction 3D décrite au chapitre précédent. Puisque la nature du mouvement de la caméra affecte la précision des résultats, nous nous sommes particulièrement intéressés à cet aspect critique de la vision dynamique.

Nous avons montré qu'en contrôlant les mouvements de la caméra à l'aide de lois de commande en boucle fermée sur la vision, on pouvait obtenir de nettes améliorations dans la reconstruction et la localisation des primitives. Les résultats obtenus en simulation confirment ainsi le point de vue de travaux antérieurs sur l'avenir très prometteur de la vision active [BAN86] [ALO87] [BAJ88] [SAN90].

L'implémentation de notre approche sur une cellule expérimentale de vision dynamique active constituée d'une caméra embarquée sur un robot à six degrés de liberté fait l'objet du chapitre 5. Les résultats concernant la reconstruction d'un environnement similaire à celui d'une zone encombrée de nombreux tuyaux d'une centrale nucléaire y sont présentés et commentés. Ces résultats viendront ainsi appuyer ceux déjà obtenus en simulation.

Dans le chapitre suivant, nous allons nous focaliser sur les aspects de traitement d'images, sans lesquels notre approche de reconstruction 3D ne pourrait fonctionner en environnement réel. Nous aborderons en effet, le problème délicat de mise en correspondance de primitives bidimensionnelles dans une séquences d'images. Une méthode robuste et efficace parfaitement adaptée à nos besoins y est présentée.

Chapitre 4

Appariement d'indices bidimensionnels

Dans ce chapitre, après avoir décrit les principales approches connues pour la mise en correspondance d'indices bidimensionnels dans une séquence d'images, nous présentons une approche locale de suivi de segments. Cette méthode originale s'appuie sur un algorithme de calcul d'éléments de contour en mouvement. Les propriétés locales de cette approche ont permis une implémentation aisée sur une machine parallèle et sur une machine séquentielle conduisant à des performances temps-réel. Enfin, la validation de cette méthode sur plusieurs séquences d'images est montrée à la fin de ce chapitre.

4.1 Introduction

L'application visée par notre travail étant la navigation et l'inspection à l'intérieur d'une centrale nucléaire souvent encombrée de nombreux tuyaux, il paraît naturel de s'intéresser particulièrement aux primitives cylindriques. Nous avons montré au chapitre précédent que des mouvements optimaux, engendrés par des lois de commande en boucle fermée sur les informations visuelles, permettaient de positionner la caméra de manière à optimiser la reconstruction 3D du cylindre. La première tâche que nous devons donc effectuer est de se focaliser sur un cylindre (ou plus exactement sur un ou deux limbes du cylindre) de la scène. Ensuite, pour que notre algorithme de reconstruction 3D puisse fonctionner, il est indispensable de suivre les limbes du cylindre observés dans la séquence d'images. Ces limbes seront perçus comme des segments de droite correspondant à des contours d'intensité.

Nous nous posons donc ici deux questions essentielles auxquelles nous tenterons d'apporter une réponse :

- (1) comment se focaliser, dans la première image de la séquence, sur un limbe parmi les limbes des cylindres présents dans la scène ?
- (2) une fois ce limbe sélectionné, comment l'apparier de manière fiable à ses correspondants dans les images suivantes ?

Il est important de souligner ici, avant de rentrer dans les détails de notre approche, que l'élément essentiel pour guider nos choix algorithmiques, sans toutefois négliger les contraintes de robustesse et de fiabilité, est l'aspect temps réel. En effet, nos algorithmes de traitement d'image doivent respecter des contraintes fortes de temps puisque une commande en boucle fermée sur la vision est à la base de notre méthode de reconstruction 3D. Une partie importante du travail a donc consisté à implanter les algorithmes choisis sur la machine parallèle Sympati2 [BAS88]. Il s'agit d'une carte multi-processeurs de type SIMD adaptée au traitement d'images, développée par le CEA en collaboration avec l'IRIT¹ et commercialisée par Centralp Automatismes. Nous reviendrons plus en détails sur cette carte dans l'annexe C.

4.2 Focalisation sur un segment de la scène

4.2.1 Introduction

Le traitement des images permet d'extraire de l'image brute, considérée comme une matrice de pixels, l'information pertinente sous forme de contours, de régions,... selon les besoins de l'application envisagée. En ce qui nous concerne, on cherche à détecter une entité image de type segment de contour ou, plus exactement, on se limitera à une chaîne de points contours. La sélection d'une chaîne potentielle est effectuée selon un critère très simple : la longueur des chaînes. Nous allons traiter ce problème en trois phases :

- (1) la détection des points de contraste.
- (2) la sélection d'une droite support significative par la transformée de Hough.
- (3) le repérage de la chaîne de points contours la plus longue sur la direction sélectionnée.

¹Institut de recherche en Informatique de Toulouse.

4.2.2 Détection des points de contraste

Les pixels image correspondant à des variations locales spatiales de niveaux de gris importantes sont appelés les points de contraste. Des recherches récentes ont mis au point des détecteurs de contours très performants tels que par exemple le filtre de Deriche [DER87] ou le filtre de Shen-Castan [SHE92], ces deux filtres ayant des performances comparables. Notre choix s'est porté sur le filtre de Shen-Castan car il est facilement parallélisable sur une machine de type SIMD telle que Sympati2. Ce filtre récursif a une fonction de transfert qui présente une discontinuité et changement de signe à l'origine. Il est optimal au regard de trois critères :

- il maximise le rapport signal/bruit;
- il localise au mieux le point de contraste;
- il aboutit à un pic unique.

4.2.3 Transformée de Hough

La transformée de Hough (TH) permet de détecter dans des images la présence de courbes paramétriques appartenant à une famille connue à partir d'un ensemble de points sélectionnés [DUD72], [MAI85]. La mise en œuvre usuelle de la TH dans ce contexte utilise la position dans l'image des points de contour, mais, parfois, tient compte également de la valeur de luminance en ces mêmes points.

4.2.3.1 Définition

Pour une famille de courbes de la forme $g(\underline{X}, \underline{P}) = 0$, on quantifie l'espace des paramètres $\{\underline{P}\}$ et on associe un accumulateur $A(\underline{P})$ à chaque élément de l'espace (cellule) ainsi quantifié.

Pour chaque point sélectionné \underline{X}_i , on détermine dans l'espace quantifié l'ensemble $\{\underline{P}\}$ de tous les paramètres possibles \underline{P}_j qui vérifient la relation :

$$g(\underline{X}_i, \underline{P}_j) = 0 \quad (4.1)$$

et on incrémente les accumulateurs correspondant aux paramètres \underline{P}_j (étape de vote) :

$$A(\underline{P}_j) = A(\underline{P}_j) + 1, \quad \underline{P}_j \in \{\underline{P}\} \quad (4.2)$$

Les meilleurs paramètres \underline{P}^* sont obtenus par la recherche des maxima locaux dans le tableau $A(\underline{P})$, soit :

$$\{\underline{P}^*\} = \underset{\underline{P}}{\text{Argmax}} A(\underline{P}) \quad (4.3)$$

4.2.3.2 Mise en œuvre de la TH

Pour la sélection d'une droite support dans l'image, nous avons choisi la transformée de Hough car elle se prête aisément à une parallélisation sur une architecture du type SIMD. En effet, chaque opération que nous avons décrite ci-dessus est répétée identiquement sur tout point M considéré dans l'image. D'autre part, c'est une méthode robuste au bruit et aux occlusions partielles.

Si on choisit la représentation (ρ, θ) des droites, l'espace des paramètres sera alors l'espace (ρ, θ) avec $0 \leq \rho \leq \Delta/2$ où Δ est la diagonale de l'image et $0 \leq \theta < 2\pi$. Cet espace est discrétisé en cellules uniformes avec un pas sélectionné de façon expérimentale. La TH que nous avons mise en œuvre procède en deux étapes, toujours dans l'optique d'optimiser le temps de calcul :

- une étape grossière avec un pas de 10 degrés pour θ et un pas de 6 pixels pour ρ . Cette étape permet de repérer la partie de l'espace de paramètres la plus dense en points de contour. Une fois la procédure de vote effectuée en utilisant les points de contraste de l'image, le maxima $(\rho_{max}, \theta_{max})$ est repéré par la cellule du tableau $A(\underline{P})$ ayant accumulé le maximum de votes.
- une étape fine avec un pas de 1 degré pour θ et de 1 pixel pour ρ . Une discrétisation plus fine de l'espace des paramètres n'est pas nécessaire car cette précision est suffisante pour explorer, par l'opération ultérieure de chaînage, la droite support qui sera sélectionnée. Dans cette seconde étape, il suffit de se focaliser sur une fenêtre autour de la valeur $(\rho_{max}, \theta_{max})$ sélectionnée à l'étape précédente. La recherche est ainsi beaucoup moins exhaustive puisque guidée par le résultat de l'étape grossière. Cette fois, le maxima $(\tilde{\rho}, \tilde{\theta})$ du tableau $A(\underline{P})$ correspond aux paramètres de la droite support sélectionnée sur laquelle il nous reste à détecter la chaîne de points contours la plus longue.

On peut aussi noter qu'il est possible de retenir d'autres critères de sélection que le maximum de votes de la TH ou de considérer, si souhaité, des contraintes particulières, comme par exemple que θ soit compris entre deux valeurs données θ_1

et θ_2 . La TH offre un moyen très simple de les mettre en œuvre grâce à la matrice d'accumulation.

4.2.4 Extraction des chaînes de points de contraste

Le processus de chaînage permet de regrouper les points de contraste en lignes de contraste. Ces lignes représentent les lieux de l'image où l'intensité lumineuse varie fortement dans une direction. Chaque ligne correspond dans la scène à une arête d'un objet, une ligne de discontinuité de surface, ou une ligne de changement de luminance (marquages, ombres, etc). Dans le cas particulier de notre application, elles représentent plus exactement les limbes des cylindres (ou parfois des ombres générées par ces derniers).

L'algorithme de chaînage que nous avons développé est relativement simple. En effet, à l'issue de l'étape précédente de classification par la transformée de Hough, nous disposons d'une information clef sur les points de contraste à chaîner : les paramètres $(\tilde{\rho}, \tilde{\theta})$ de la droite support sélectionnée. Le chaînage s'effectue en trois étapes :

- (1) création des chaînes : la recherche des points d'une chaîne s'effectue dans un voisinage 8-connexe, le champ de recherche étant limité à une bande centrée sur la droite support définie par les paramètres $(\tilde{\rho}, \tilde{\theta})$;
- (2) élimination des chaînes trop courtes et fusion des chaînes voisines ;
- (3) sélection de la chaîne la plus longue : elle représente le "segment virtuel" qu'il faudra suivre tout au long de la séquence d'images. En effet, comme cela sera motivé ci-dessous, on ne procède pas à une approximation polygonale des chaînes de contour et on s'en tient aux chaînes de points contours comme niveau de représentation.

Notons que d'autres critères que la longueur des chaînes peuvent évidemment être considérés pour le choix d'une chaîne potentielle. Par ailleurs, si la chaîne sélectionnée n'est pas satisfaisante (longueur inférieure à un seuil prédéfini), le processus de chaînage est relancé en considérant le maxima suivant de la TH.

4.3 Mise en correspondance de segments dans une séquence d'images

4.3.1 Introduction

La mise en correspondance d'indices bi-dimensionnels entre deux images successives est loin d'être un problème trivial. En effet, autant il est facile pour un être humain d'effectuer cette mise en correspondance, autant l'élaboration d'un algorithme permettant de la réaliser de manière automatique, fiable et robuste est une tâche ardue [SHE87], [MED84], [CHE90], [DER90], [XIE89], [FLA91], [AYA89a]. La complexité de cette mise en correspondance vient du nombre généralement élevé d'indices dans les images et des ambiguïtés possibles d'appariement.

La définition d'un algorithme de mise en correspondance repose sur deux choix fondamentaux :

- les primitives extraites de l'image que l'on met en correspondance;
- les contraintes permettant de dégager un ensemble d'appariements corrects.

4.3.2 Choix des primitives

Pour effectuer la mise en correspondance, il est nécessaire au préalable d'extraire des deux images un ensemble d'indices bidimensionnels (ou primitives) qui se distinguent clairement de leur voisinage :

- (1) Les points caractéristiques : ce sont les points de l'image présentant localement une forte variation de la luminance. Le choix de ces primitives conduit à des représentations 3D pauvres.
- (2) Les points de contour : cette représentation permet une modélisation beaucoup plus riche que la précédente.
- (3) Les régions : elles portent une information intrinsèque très riche (paramètres de surface, forme, frontière, etc). Cependant, les diverses techniques de segmentation en régions existantes sont en pratique assez instables.

Par conséquent, la représentation par points de contours est généralement la plus appropriée. A partir de ces points, des représentations dérivées plus élaborées sont

considérées : approximation de lignes de contraste par des primitives géométriques de plus haut niveau (segments, arcs d'ellipse, jonctions, etc).

Le choix d'en rester au niveau pixel ou de passer aux représentations dérivées est justifié principalement par les arguments suivants :

- les représentations “segments” permettent, lorsqu'elles sont pertinentes dans le type d'environnement considéré, de limiter la combinatoire par la réduction du nombre total de primitives et l'enrichissement intrinsèque portée par chacune d'entre elles [MED84], [XIE89], [DER90], [AYA89a]. Cependant, leur extraction passe par une phase de segmentation (approximation polygonale des contours) qui s'avère instable. Notamment, des segments peuvent être fragmentés d'une image à l'autre ou n'être pas identiquement extraits (perte d'une partie du segment dans l'une des deux images), ce qui ne facilite pas la tâche de mise en correspondance déjà délicate car les paramètres globaux du segment (longueur, orientation) en sont évidemment fortement perturbés.
- les représentations “points” moins riches que les précédentes s'avèrent plus simples à manipuler [SHE87], [CHE90], [FLA91]. En particulier, les algorithmes de mise en correspondance basés sur les points se prêtent nettement mieux que ceux basés sur les segments à une implémentation temps-réel. Cependant, ces algorithmes s'avèrent moins fiables que les précédents car les primitives points sont des primitives moins pertinentes que les segments.

En ce qui nous concerne, nous avons choisi un compromis entre les deux types de primitives. Nous utiliserons en effet des chaînes de point de contour qui sont des primitives séduisantes par le fait qu'elles portent la même richesse intrinsèque que les segments tout en gardant les propriétés locales d'une représentation pixel. Cette représentation, outre sa robustesse, va nous permettre de porter aisément notre algorithme de mise en correspondance sur une machine parallèle.

4.3.3 Méthodes de mise en correspondance

Différentes méthodes ont été répertoriées dans la littérature [MAR82]. Nous présentons brièvement une revue des principales méthodes connues.

4.3.3.1 La corrélation

Le critère de mise en correspondance de deux points est la recherche de la meilleure corrélation entre des régions centrées autour de ces points, appliquée à des images

de niveau de gris ou à des images binaires après transformation (seuillage, Laplacien,...). Ces dernières conduisent à des pics de corrélation plus marqués [SHE87]. La corrélation, lorsqu'elle est utilisée sur des points d'intérêt, fournit des cartes très peu denses. Par contre, utilisée sur l'ensemble de l'image, on obtient des cartes denses mais au prix d'un effort de calcul très important.

L'intérêt de cette approche réside essentiellement dans sa simplicité de mise en œuvre. Il faut cependant noter que la mesure de corrélation est perturbée par des phénomènes d'occlusion ainsi que par des transformations géométriques non "translationnelles".

4.3.3.2 La programmation dynamique

L'idée de base dans la programmation dynamique est de remplacer l'optimisation d'une fonction multivariable par la résolution d'un certain nombre de problèmes d'optimisation plus simples. Le problème de la recherche d'appariements est posé comme une recherche de chemin optimal dans un graphe. Chaque noeud du graphe correspond à la mise en correspondance de deux primitives. Un arc relie deux couples qui semblent simultanément associables. Le chemin optimal est déterminé selon un critère de similarité locale des primitives appariées et un critère de cohérence entre primitives voisines (par exemple les pixels appartenant à une même chaîne de contours) en se basant sur une fonction de coût. Un exemple de ce type d'algorithme basé sur des primitives point est décrit dans [FLA91].

Ces méthodes complexes donnent en général de bons résultats mais au prix de calculs très coûteux.

4.3.3.3 La relaxation

De nombreux algorithmes d'appariement utilisent des méthodes de relaxation. Ces algorithmes consistent à définir initialement des hypothèses d'appariement, chacune d'elles étant affectée d'un certain poids caractérisant la vraisemblance de l'appariement. Le poids de ces différents appariements évolue alors selon un processus itératif de propagation prenant en compte des contraintes locales entre voisins. L'information de contexte joue ici un rôle prépondérant. Un appariement entre deux primitives sera validé s'il est cohérent dans son contexte.

Différentes versions de cette approche générale ont été proposées dans la littérature notamment pour l'appariement de points [MOU89] ou de droites [LON86], [MED84]. Par ailleurs, des méthodes d'étiquetage hiérarchique s'avèrent plus intéres-

santes. Une première étape d'appariement (simple et pas très coûteuse) est utilisée pour “dégrossir” la recherche, permettant une analyse plus rapide. La seconde étape, plus élaborée permet d'affiner le processus d'appariement [WAN90].

Ces algorithmes donnent en général de très bons résultats. Leur principal inconvénient est leur caractère exhaustif. En effet, ils nécessitent de générer au départ l'ensemble des appariements potentiels et de mettre à jour à chaque étape de relaxation l'ensemble des valeurs de vraisemblance associées.

4.3.3.4 Méthodes de prédiction-vérification d'hypothèses

Ces méthodes consistent dans une première étape, à sélectionner un nombre réduit de segments uniformément répartis dans une image. Pour chacun, on effectue une recherche exhaustive de ses correspondants potentiels. Chaque couple trouvé constitue une hypothèse [RDE86], [AYA89a], [BOU89a]. La seconde étape consiste à tester chacune de ces hypothèses. Etant donné une hypothèse (S_i^1, S_j^2) , on cherche à apparier chacun des voisins de S_i^1 . En cas de succès, on relance la propagation à partir du nouveau segment obtenu jusqu'à ce qu'un nouvel appariement ne puisse plus être déduit. Toutes les primitives appariées concourent à l'augmentation du score associé à chaque appariement. Notons qu'au cours de ce processus, on utilise implicitement la contrainte de continuité.

Toutes les hypothèses ayant été propagées, on retient pour chaque segment l'appariement qui a reçu le meilleur score c'est à dire, qui présente le plus grand nombre d'appariements consistants avec lui. Le coût d'évaluation des hypothèses est assez élevé. Cependant, cette méthode présente l'avantage d'être fortement parallélisable car toutes les hypothèses peuvent être propagées indépendamment.

4.3.3.5 Méthodes récursives

Ces méthodes, plus communément connues sous le nom de “token tracker”, sont basées sur un filtre récursif, le filtre de Kalman. Ces approches ont déjà été évoquées dans le premier chapitre consacré au problème du calcul de la structure 3D à partir du mouvement. Il est en effet difficile de découpler le problème de l'estimation du mouvement et celui de l'appariement dans ces approches récursives car ils sont très intimement liés.

En général, les primitives ou “tokens” à suivre dans la séquence d'images sont des segments 2D [CRO92], [DER90] ou même des segments 3D [ZHA90]. Dans ces approches, contrairement aux approches précédentes, une cohérence temporelle est

utilisée pour assurer le suivi de ces primitives. En effet, les primitives sont non seulement définies par leurs paramètres structurels mais aussi par leurs paramètres cinématiques ainsi que les matrices de covariance associées.

Trois étapes essentielles régissent le processus d'appariement : une étape de prédiction, une étape d'appariement (modèles-observations) et enfin une étape de mise à jour. Ces approches permettent d'affiner une estimation a priori de la position des primitives dans la scène. Cela permet de déduire un certain nombre d'appariements de primitives qui, eux mêmes contribuent à affiner l'estimation de position et donc permettent de nouveaux appariements de primitives.

4.3.4 Conclusion

Les méthodes de mise en correspondance basées sur les points ne nous paraissent pas en mesure de faire face aux multiples problèmes rencontrés en pratique (bruit, occlusion, etc). Les autres méthodes, plus sûres, sont toutes basées sur l'extraction explicite des segments de contour après chaînage et approximation polygonale des chaînes de contour obtenues. Le critère d'appariement met en jeu des paramètres globaux des segments comme leur longueur et leur orientation, parfois leur contraste moyen. Ces techniques sont assez performantes, mais elles sont par contre sensibles à l'instabilité de la phase de segmentation (fragmentation d'un segment d'une image à l'autre). De plus, elles requièrent l'élaboration d'une carte dédiée pour obtenir une solution temps réel (quand celle-ci peut être obtenue).

Nous avons préféré privilégier une approche locale qui, en fait, ne manipule jamais explicitement de segments de contour [BOU92a], [BOU93b], [BOU93c], [BOU93a]. Elle ne nécessite la recherche de correspondants aux points de la chaîne de contour que dans une seule direction. Elle présente les avantages suivants :

- une seule structure de données est considérée : la matrice de pixels ;
- elle est implémentable sur des cartes programmables offrant des opérations de convolution temps réel ;
- elle est robuste aux aléas de segmentation et d'occlusion ;
- elle est adaptée au niveau d'information requis par les étapes ultérieures de la reconstruction 3D tout en étant générale et versatile.

La méthode de mise en correspondance de segments que nous avons mise en œuvre s'appuie sur un algorithme de calcul d'éléments de contours en mouvement (ECM) précédemment développé à l'IRISA [BOU87], [BOU89b]. Cet algorithme a déjà été intégré dans un processus de mise en correspondance de segments [XIE89] mais uniquement pour *prédire* la position dans les images suivantes de segments *explicites* extraits par une approximation polygonale des contours. Avant de décrire notre méthode qui représente une extension significative de cet algorithme, nous allons en rappeler brièvement les principes de fonctionnement.

4.3.5 Détermination des appariements locaux de points de contour

Cette méthode consiste en une modélisation spatio-temporelle d'un élément de contour en mouvement et utilise un test de rapport de vraisemblance pour en déterminer la localisation, l'orientation spatiale et la composante de déplacement perpendiculaire au contour.

4.3.5.1 Modélisation d'un élément de contour en mouvement (ECM)

Une séquence d'images est considérée comme un espace à trois dimensions : deux dimensions spatiales (X, Y) et une dimension temporelle t . Dans cet espace, un élément de contour en mouvement (ECM) correspond à une portion de surface spatio-temporelle 3D. Cette surface peut être approximée localement par une portion de surface plane avec comme représentation paramétrique (voir la figure 4.1) :

- (1) une position (X_0, Y_0, t_0) et une direction spatiale θ de l'élément de contour ;
- (2) un vecteur de déplacement \vec{V} associé à l'ECM.

En fait, la représentation paramétrique du plan est plus exactement sa position (X_0, Y_0, t_0) et deux angles θ et ψ (par rapport à l'axe t), que l'on peut ensuite de façon évidente relier aux paramètres des ECM. On retrouve (X_0, Y_0, t_0) , position de l'ECM et θ son orientation spatiale dans le plan image, et la composante perpendiculaire de vitesse est donnée par $V^\perp = \tan \psi$.

Le choix d'une telle approximation se justifie par le fait que l'on considère un écart de temps très faible entre deux images. Des modèles de surface plus complexes peuvent également être considérés [MAR87]. La méthode décrite ici permet

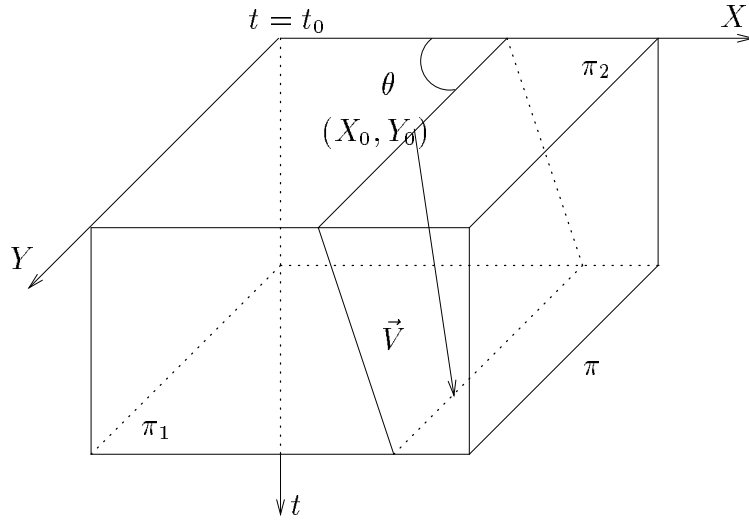


Figure 4.1 : Modélisation géométrique d'un ECM.

la détection de telles portions de plan et l'estimation de leurs paramètres θ et V^\perp . Rappelons que seule la composante orthogonale V^\perp de \vec{V} pourra être déterminée, la composante tangentielle étant impossible à mesurer dans une fenêtre d'observation locale [BOU87], [BOU89b].

Soit π le volume placé au point (X_0, Y_0, t_0) dans l'espace (X, Y, t) . On prend en compte les deux hypothèses de configuration locale suivantes :

(1) **Hypothèse H_0 :**

Il n'existe pas d'ECM dans le volume π . La distribution des intensités lumineuses à l'intérieur du volume π peut être modélisée par un niveau de gris constant c_0 perturbé par un bruit gaussien centré de variance σ^2 , soit : $\{c_0, \sigma^2\}$.

(2) **Hypothèse H_1 :**

Il existe un ECM dans le volume π . Cela se traduit par la présence d'une portion de plan divisant le plan π en deux sous volumes π_1 et π_2 . Comme dans le cas précédent, la distribution des intensités lumineuses dans ces deux volumes est modélisée par :

- $\{c_1, \sigma^2\}$ pour le sous-volume π_1 .
- $\{c_2, \sigma^2\}$ pour le sous-volume π_2 avec $c_1 \neq c_2$.

4.3.5.2 Test de vraisemblance pour la détermination des ECM :

La distribution des intensités lumineuses dans un volume π suit un modèle de densité Gaussienne :

- $N(c_0, \sigma^2)$ en tout point du volume π pour l'hypothèse H_0 .
- $N(c_1, \sigma^2)$, $N(c_2, \sigma^2)$ respectivement en tout point du volume π_1 et du volume π_2 pour l'hypothèse H_2 .

A chaque hypothèse peut être associée une fonction de vraisemblance, densité jointe des variables considérées. Ces dernières seront supposées indépendantes.

- Pour l'hypothèse H_0 , la fonction de vraisemblance L_0 s'exprime par :

$$L_0 = (2\pi\sigma^2)^{-\frac{n}{2}} \prod_{p_i \in \pi; i=1}^n \exp\left\{-\frac{1}{2\sigma^2}(f_{p_i} - c_0)^2\right\} \quad (4.4)$$

- Pour l'hypothèse H_1 , la fonction de vraisemblance s'écrit :

$$L_1 = (2\pi\sigma^2)^{-\frac{n_1+n_2}{2}} \prod_{p_j \in \pi_1; j=1}^{n_1} \exp\left\{-\frac{1}{2\sigma^2}(f_{p_j} - c_1)^2\right\} \prod_{p_k \in \pi_2; k=1}^{n_2} \exp\left\{-\frac{1}{2\sigma^2}(f_{p_k} - c_2)^2\right\} \quad (4.5)$$

où :

- f_m est l'intensité lumineuse du point m .
- n, n_1 et n_2 représentent respectivement le nombre de points dans le volume π, π_1 et π_2 .

Un ECM est déterminé à l'aide d'un test de rapport de vraisemblance. La sélection de l'hypothèse H_1 contre H_0 s'appuie sur l'expression suivante :

$$\xi = \ln(L_1/L_0) \quad (4.6)$$

qui dépend de deux types de paramètres :

- les paramètres géométriques $\Phi = (X_0, Y_0, \theta, V^\perp)$ qui déterminent la portion de plan modélisant un ECM.

- les paramètres photométriques et statistiques notés α qui déterminent les valeurs moyennes (c_0, c_1, c_2) et la variance du bruit σ^2 sur les intensités lumineuses.

L'hypothèse H_1 est retenue si la valeur maximale du rapport $\xi(\Phi, \alpha)$ est supérieure à un seuil prédéterminé :

$$\max_{\Phi} \max_{c_1, c_2} \min_{c_0} \xi(\Phi, \alpha) \geq \lambda \quad (4.7)$$

4.3.5.3 Dérivation du critère sous forme d'une convolution :

Considérons une géométrie Φ_g prédéfinie, dans le domaine de variation possible des paramètres θ et V^\perp . Après évaluation des estimateurs optimaux de c_0, c_1, c_2 , la racine carrée de la valeur maximale de la fonction ξ peut s'écrire [BOU87], [BOU89b] :

$$\zeta(\Phi_g) = \left| \sum_{t=t_1}^{t_2} \sum_{\eta_t \in M_t} a_{\eta_t}^g f(p + \eta_t) \right| \quad (4.8)$$

avec M_t un jeu d'indices tel que $(p + \eta_t)$ désigne pour le plan image t tous les points appartenant au volume π centré au point P et où les coefficients $a_{\eta_t}^g$ sont donnés par :

$$\begin{cases} a_{\eta_t}^g = \frac{1}{\sigma} \left(\frac{n_2}{2n_1(n_1+n_2)} \right)^{1/2}, & \text{si } \eta_t \in \pi_1 \\ a_{\eta_t}^g = -\frac{1}{\sigma} \left(\frac{n_1}{2n_2(n_1+n_2)} \right)^{1/2}, & \text{si } \eta_t \in \pi_2 \end{cases} \quad (4.9)$$

Le critère de vraisemblance devient alors :

$$\max_{\Phi_g} \zeta(\Phi_g) \geq \lambda \quad (4.10)$$

Ainsi, le calcul du maximum de vraisemblance est réduit à une simple opération de convolution de la fonction intensité avec un jeu de masques associés à une géométrie prédéfinie, qu'on peut donc calculer hors-ligne.

Par ailleurs, une contrainte supplémentaire est introduite pour éviter les fausses détections et pour améliorer la robustesse en présence de bruit ou de forte densité des contours spatiaux. Avant de décider qu'un ECM est détecté ou pas selon le critère (4.10), la double inégalité suivante doit être vérifiée :

$$\mu_1 \leq |\zeta_{t_1}(\Phi_g) / \zeta_{t_2}(\Phi_g)| \leq \mu_2 \quad (4.11)$$

avec :

$$\zeta_t(\Phi_g) = \sum_{\eta_t \in M_t} a_{\eta_t}^g f(p + \eta_t) \quad (4.12)$$

où μ_1 et μ_2 sont deux seuils prédéterminés. Ainsi, la sélection de candidats pour lesquels les contrastes sont proches est favorisée.

Enfin, le procédé peut être optimisé en effectuant préalablement une détection de contours spatiaux, les ECM étant avant tout des éléments de contours spatiaux. On aura alors à estimer uniquement les composantes de vitesse perpendiculaires à ces points contours [XIE89]. Notons que dans cette version de base, on teste systématiquement toutes les orientations et les déplacements autorisés.

4.3.5.4 Discussion

L'intérêt de la méthode que nous venons de décrire réside dans le fait que :

- la modélisation exploite au mieux l'information sur les aspects géométrique et photométrique des pixels contour en mouvement ;
- l'opération de mesure de vraisemblance peut être ramenée à une simple opération de convolution avec des masques prédéterminés. Or, l'architecture SIMD de la machine Sympati2 est parfaitement adaptée à ce type d'opération. Cette méthode répond donc à nos objectifs temps réel.

Il faut cependant préciser que la détermination de l'ensemble des ECM d'une image pourrait être relativement lourde (hors temps réel) notamment quand la densité des contours spatiaux est importante. Or, dans le cadre de notre application, il s'agit de traiter un seul segment à la fois dans l'image, voire deux pour suivre les deux limbes d'un même cylindre, donc d'effectuer le calcul des ECM sur une petite partie des contours spatiaux de l'image. En effet, la méthode de reconstruction 3D s'effectue selon une approche de type vision active, un mouvement optimal est appliqué à la caméra et est géré en boucle fermée sur les informations capteur [BOU92b], [CHA92], [BOU93a]. Elle entraîne donc une focalisation tour à tour sur chacun des cylindres d'intérêt. Cela réduit considérablement la charge de calcul et va nous permettre comme on le verra par la suite, d'aboutir à des temps d'exécution tout à fait raisonnables.

4.3.6 Application au suivi d'un seul segment

Une fois l'étape de focalisation effectuée, il nous faut mettre en correspondance et suivre le segment "virtuel" sélectionné dans la séquence d'images. Nous allons proposer ici une méthode originale qui exploite une version appropriée de l'algorithme de détermination des ECM. C'est de ce fait, une méthode locale puisqu'opérant au niveau pixel. Elle consiste en les étapes suivantes :

- calcul des masques de convolution.
- détermination de la zone de recherche.
- calcul du test de vraisemblance.

que nous décrivons à présent.

4.3.6.1 Calcul des ECM

Etant donné la liste de pixels L_i représentant un segment virtuel S_i d'une image i de la séquence, on cherche à déterminer les correspondants de ces points dans l'image suivante $i+1$. L'algorithme que nous avons mis en œuvre procède de la manière suivante :

1. Masque de convolution optimal : La connaissance de l'orientation $\hat{\theta}$ du segment de l'image i (en fait celle de la droite support calculée après moindres carrés comme expliqué plus loin) permet de choisir les masques de convolution les mieux adaptés pour la mesure de vraisemblance. Ces masques de convolution optimaux garantissent l'obtention de résultats nettement plus performants que la version de base. Rappelons que dans [BOU89b], il est proposé de considérer un jeu de masques correspondant à une quantification assez grossière (par exemple tous les 45°). Dans notre cas, les masques de convolution correspondant à une direction donnée θ , $0^\circ \leq \theta \leq 180^\circ$, sont calculés avec une précision de 1° , ce qui est tout à fait satisfaisant pour la détermination du déplacement 2D apparent.

Soit θ^* , la valeur entière la plus proche de $\hat{\theta}$, pour laquelle on calcule les masques de convolution correspondants $M_{|\theta^*}^1$ et $M_{|\theta^*}^2$ à partir de (4.9). L'image i (respectivement l'image $i+1$) est alors convoluée avec le masque optimal $M_{|\theta^*}^1$ (respectivement $M_{|\theta^*}^2$). En fait, on a $M_{|\theta^*}^1 = M_{|\theta^*}^2$ puisque l'on ne considère que des déplacements entiers [BOU89b].

2. Espace de recherche : Pour tout point contour de l'image i , on recherche, dans un espace prédéfini de l'image $i+1$, le point correspondant pour lequel la mesure de vraisemblance est maximale. La recherche la plus efficace doit s'effectuer dans la direction orthogonale au contour (voir la figure 4.2).

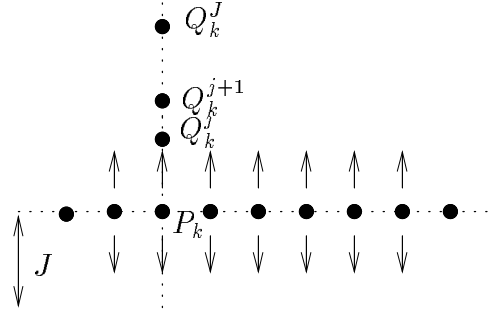


Figure 4.2 : Recherche des ECMs dans la direction orthogonale au contour.

où les notations introduites ont les significations suivantes :

- $P_k \in L_i$: point contour du segment à l'instant t .
- Q_k^j : nouvelle position possible du point P_k dans la direction orthogonale au contour à l'instant $t+1$.
- $P_k \xrightarrow{?} Q_k^j$, $j \in [-J, J]$: appariement possible entre les points P_k à l'instant t et le point Q_k^j à l'instant $t+1$, avec J l'amplitude maximale des déplacements recherchés.

Selon l'orientation de la normale à P_k , les déplacements ne correspondent pas toujours à des positions Q_k^j entières de la matrice de pixels (problèmes d'échantillonnage). L'évaluation en des positions non entières nécessite soit de recalculer les masques de convolution, soit de procéder à des interpolations sur la fonction intensité. Ces opérations sont trop coûteuses. Aussi avons nous opté pour une solution qui, d'une part, satisfait les contraintes temps réel, et d'autre part, ne manque pas d'efficacité quant à la recherche des ECM. En effet, nous avons pris en considération quatre directions possibles de recherche : $\delta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ qui, elles, induisent des positions Q_k^j entières après déplacement (voir la figure 4.3). Parmi ces quatre directions, on sélectionne la direction δ^* la plus proche de la direction orthogonale à θ^* :

$$\delta^* = \delta_i / \delta_i \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \text{ et } \left| (\theta^* + \frac{\pi}{2}) - \delta_i \right| \pmod{\pi} < \frac{\pi}{4} \quad (4.13)$$

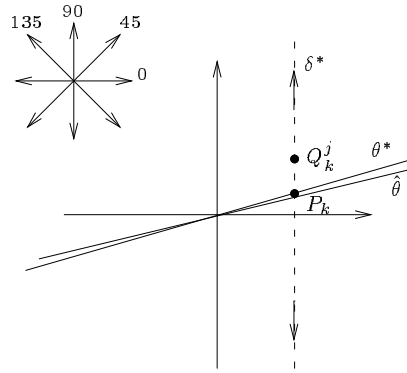


Figure 4.3 : Choix de la direction de recherche

Il faut souligner que l'objectif est d'apparier deux positions successives d'un limbe d'un cylindre. On exploite ici le fait que l'on n'a nul besoin de connaître le déplacement 2D complet de chaque élément de contour. Seule nous intéresse in fine la détermination des nouvelles valeurs des paramètres ρ, θ de la projection du limbe du cylindre dans l'image $i+1$.

La recherche des correspondants des points P_k de la liste L_i est alors effectuée dans la direction δ^* , l'amplitude maximale des déplacements recherchés étant prédéfinie. Si l'on peut prédire la prochaine position du segment, la zone de recherche peut alors être délimitée autour de cette prédiction. Dans le cas de notre approche en boucle fermée, cette information se déduit aisément de la structure 3D estimée du cylindre correspondant [BOU92b]. De plus, la zone de recherche sera de plus en plus limitée au fur et à mesure de la convergence du processus.

3. Détermination des ECM selon le test de vraisemblance : Pour chaque point P_k de la liste L_i de l'image i , étant donné un déplacement $P_k Q_k^j$ dans la direction de recherche δ^* , la valeur de vraisemblance ζ est donnée par la valeur absolue de la somme des deux valeurs issues de la convolution des images i et $i+1$ avec le masque $M_{|\theta^*}^1$ aux points P_k et Q_k^j respectivement. Le point correspondant à la valeur maximale $\hat{\zeta}$, à condition qu'elle soit supérieure à un seuil λ prédéfini, est enregistré dans la liste L_{i+1} . Notons que avant de décider qu'un ECM a été trouvé, il doit vérifier aussi la relation (4.11) qui favorise les candidats de contrastes proches. On obtient ainsi une nouvelle liste de points qui sont les homologues des points de la liste précédente L_i .

4.3.6.2 Estimation des paramètres de la droite support des ECM

Connaissant la liste L_{i+1} des ECM ou, plus exactement, des correspondants locaux aux points de la liste L_i , nous utilisons une technique de moindres carrés pour estimer les paramètres $(\hat{\rho}_{i+1}, \hat{\theta}_{i+1})$ de la droite support dans l'image $i+1$. Ensuite, on calcule les résidus du moindre carré qui vont constituer une mesure de la qualité de l'estimation :

$$\varepsilon_{quad} = \frac{1}{n} \sum_{j \in L_{i+1}} (x_j \cos \hat{\theta}_{i+1} + y_j \sin \hat{\theta}_{i+1} - \hat{\rho}_{i+1})^2 \quad (4.14)$$

n étant le nombre de points de la liste L_{i+1} .

Si l'erreur ε_{quad} est importante, il s'agit d'un mauvais appariement. Il peut arriver en effet que l'algorithme de calcul des ECM soit en présence d'une ambiguïté très forte : si deux contours, de contrastes de valeurs très proches et de signes identiques sont présents dans le champ de recherche, la confusion est possible. En effet, certains points vont s'apparier à un contour, d'autres points avec un second contour. Ces cas rares sont observés en pratique lorsque les images présentent une symétrie de motifs répétitifs.

Par ailleurs, à cause notamment de perturbations, l'algorithme peut détecter ponctuellement quelques "faux ECM". Ces valeurs aberrantes sont éliminées en parcourant la liste de points L_{i+1} et en vérifiant qu'elles sont compatibles avec les valeurs estimées $(\hat{\rho}_{i+1}, \hat{\theta}_{i+1})$:

$$\left| x_j \cos \hat{\theta}_{i+1} + y_j \sin \hat{\theta}_{i+1} - \hat{\rho}_{i+1} \right| < \varepsilon_{aber} \quad (4.15)$$

La liste L_{i+1} est alors mise à jour et le moindre carré est relancé pour affiner l'estimation des paramètres du segment suivi. Notons qu'une "maintenance" (ajout de points) de la liste L_{i+1} s'impose également pour assurer un suivi robuste et fiable du segment virtuel S_{i+1} . En effet, il est possible que les points situés à l'une des extrémités du segment n'aient pas trouvé de correspondants. Ce phénomène est d'autant plus marqué que le mouvement de glissement du segment sur lui-même est important. D'autre part, des occultations partielles du segment peuvent aussi contribuer à "l'altération" de la liste L_{i+1} . Celle-ci doit alors être mise à jour dès que la partie occultée du segment est à nouveau visible. Enfin, il faut substituer aux quelques points Q_k^j éventuellement déclarés aberrants les véritables points situés sur le support calculé. Cette procédure de "maintenance" consiste simplement à rechercher sur la droite support calculée les points dont la valeur de convolution ξ_{t2} est proche de celle des points déjà appariés.

Notons que la méthode d'appariement décrite peut très bien s'appliquer au suivi simultané de plusieurs segments dans la séquence d'images, notamment s'ils sont parallèles, les masques de convolution étant alors identiques. Il n'y a aucune limitation intrinsèque sur ce point.

Il faut aussi souligner que pour effectuer l'appariement, on n'a en fait plus besoin de procéder à une détection des points de contour dans les images successives après t_0 . Celle-ci est quand même réalisée pour permettre la "maintenance" de la liste L_{i+1} évoquée ci-dessus (elle peut éventuellement être faite très localement).

4.3.6.3 Rappel de l'algorithme

(1) Etape de focalisation :

- (1) détection des contours spatiaux dans la première image.
- (2) sélection d'une droite significative (ρ_0^*, θ_0^*) (transformée de Hough).
- (3) repérage du segment le plus long sur cette direction spatiale \implies liste de pixels L_0 .

(2) Etape de suivi :

- (1) recherche des correspondants des pixels de la liste L_i dans l'image suivante (méthode des ECM) \implies liste L_{i+1} .
- (2) estimation par moindres carrés des paramètres $(\hat{\rho}_{i+1}, \hat{\theta}_{i+1})$ de la droite support des points de la liste L_{i+1} .
- (3) mise à jour de L_{i+1} .
- (4) $i + 1 \longrightarrow i$; aller à (2.1).

4.4 Présentation des résultats

L'algorithme de suivi de segments qui vient d'être présenté a été testé sur plusieurs séquences d'images. Les différentes scènes considérées ne sont pas forcément constituées d'objets cylindriques, l'objectif étant de montrer les performances de notre approche en environnement quelconque.

Tous les tests ont été réalisés avec les paramètres suivants :

- masques de taille 3×3 .
- seuil de vraisemblance $\lambda = 1500$.
- coefficients $\mu_1 = 0.75$ et $\mu_2 = 1.25$.
- amplitude maximale des déplacements = ± 7 .

4.4.1 Séquence “table”

La scène enregistrée est constituée d’une table sur laquelle est posée une boîte blanche. La séquence a été acquise en faisant des rotations de 1° entre chaque image autour d’un axe fixe aligné avec la verticale de l’image. La boîte a été simultanément déplacée de gauche à droite. La figure (4.4) représente respectivement à gauche la première et à droite la dernière image de la séquence en niveaux de gris.

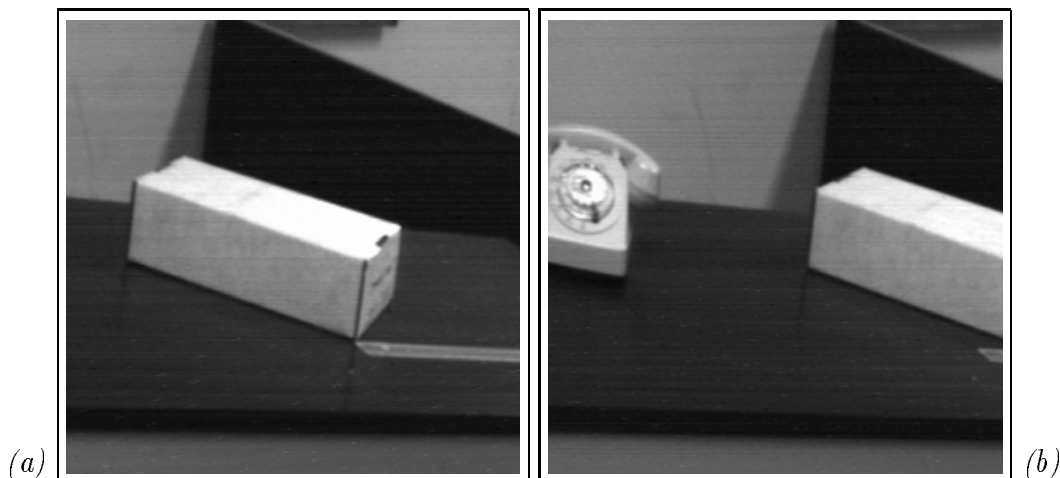


Figure 4.4 : (a) Première image de la séquence “table”; taille : 256×256 . (b) Dernière image de la séquence “table”. (Provenance : INRIA Sophia-Robovis).

La figure (4.5) montre en gras le segment sélectionné initialement par la transformée de Hough. Nous illustrons alors le suivi de ce segment dans la séquence d’images par la figure (4.6). A gauche est représentée la séquence de chaînes de points contour correspondant au segment initial sélectionné et à droite, est mise en évidence la droite support de la position finale estimée du segment en accord avec le segment extrait de cette image. On peut remarquer que le suivi du segment est parfaitement maîtrisé par notre algorithme tout au long de la séquence d’images.

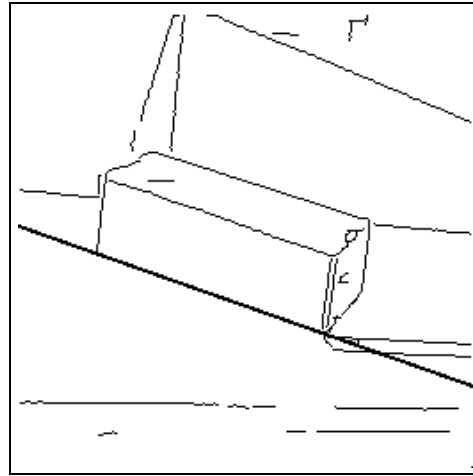


Figure 4.5 : Droite support du segment sélectionné dans la première image.

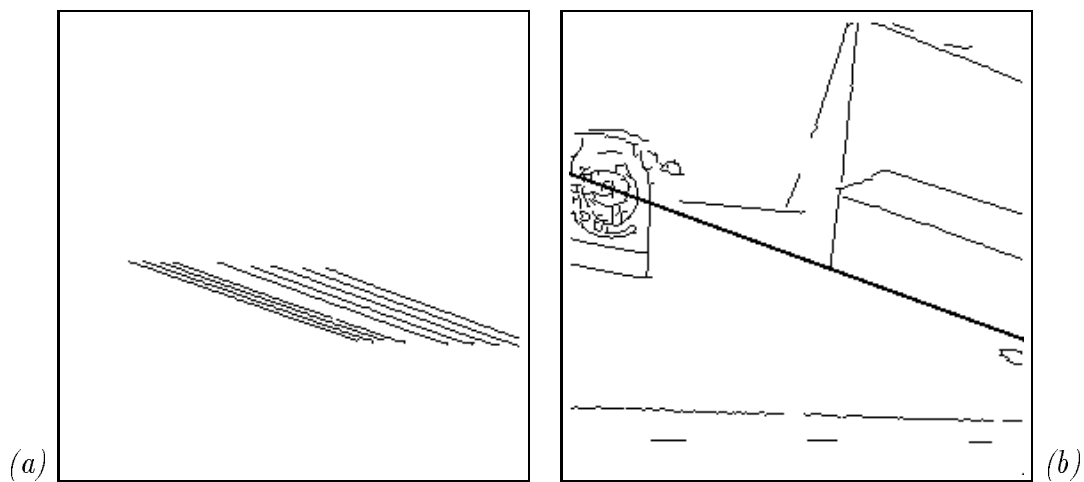


Figure 4.6 : (a) Séquence de segments correspondant au segment sélectionné. (b) Position finale du segment estimé en accord avec le segment extrait de cette image.

4.4.2 Séquence “tuyaux”

La scène dans ce cas est constituée de plusieurs tuyaux et d’une vanne. Elle illustre ainsi adéquatement l’intérieur des bâtiments d’une centrale nucléaire. Cette séquence a été acquise en faisant effectuer de petits déplacements à la caméra autour de la scène qui, elle, reste fixe. La figure (4.7) montre la première et la dernière image de la séquence en niveaux de gris. Sur la figure (4.8) est représenté en gras le segment qui a été sélectionné. Le résultat du suivi des points de ce segment est montré sur la figure (4.9) qui révèle une fois de plus l’efficacité de notre algorithme.

Notons qu’une technique basée sur la segmentation polygonale aurait difficilement suivi ce segment sans être confrontée (entre autres) à une confusion avec le second limbe du cylindre ou à des fragmentations instables de l’approximation polygonale. Par contre, notre algorithme qui prend en compte l’information photométrique des ECM différencie aisément les deux segments et tout risque de confusion est ainsi écarté. Notons aussi que durant le mouvement panoramique de la caméra à travers la scène, la projection dans l’image de la scène aux alentours du segment d’intérêt change de manière très significative (voir figure 4.9) mais cela n’affecte pas du tout notre algorithme.

De même, l’approche locale qui a été choisie montre ici tout son intérêt. En effet, réaliser l’appariement global entre les deux segments requiert seulement qu’un nombre suffisant d’appariements locaux soit effectué : donc, robustesse aux ambiguïtés locales, aux occlusions partielles et même possibilité de sous-échantillonner le nombre d’éléments de contour pris en compte dans la chaîne extraite.

4.4.3 Séquence “mire”

La scène enregistrée est constituée d’un cylindre vertical et d’une mire qui restent fixes tout au long de l’acquisition de la séquence. Entre la première et la dernière image, la caméra a effectué une translation de 500 mm dans un plan horizontal et une rotation de 35° autour de l’axe vertical de la caméra, la scène étant située à environ 1m. La figure (4.10) représente la première et la dernière image de la séquence en niveaux de gris.

La figure (4.11) montre en gras le segment sélectionné initialement. Ici la TH n’a été lancée que sur la partie droite de l’image (si l’on considère toute l’image un des segments du cylindre est évidemment sélectionné).

Comme dans le cas précédent, le résultat du suivi du segment sélectionné dans la séquence est montré sur la figure (4.12). Ici aussi notre algorithme donne d’ex-

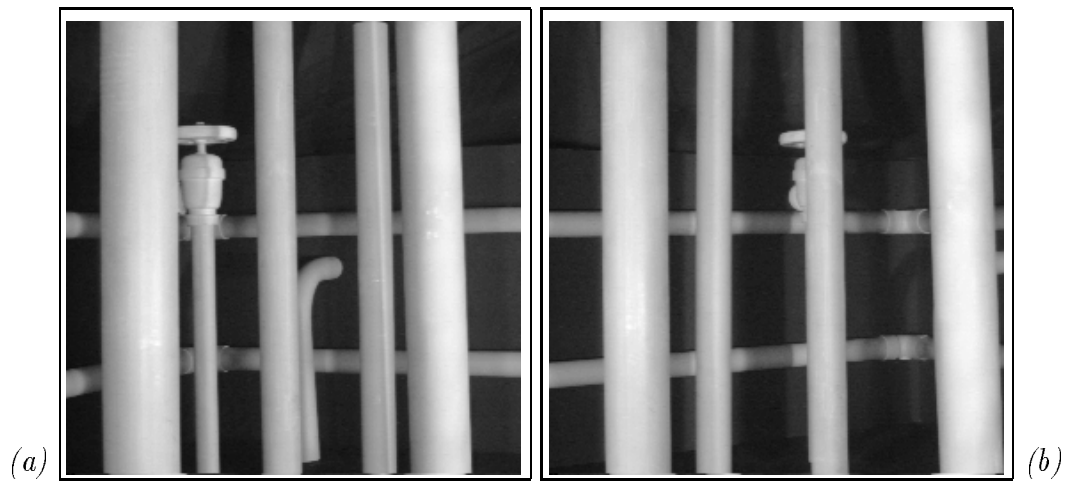


Figure 4.7 : (a) Première image de la séquence “tuyaux”, taille : 256×256 . (b) Dernière image de la séquence “tuyaux”. (Provenance : CEA-LETI-DEIN Saclay).

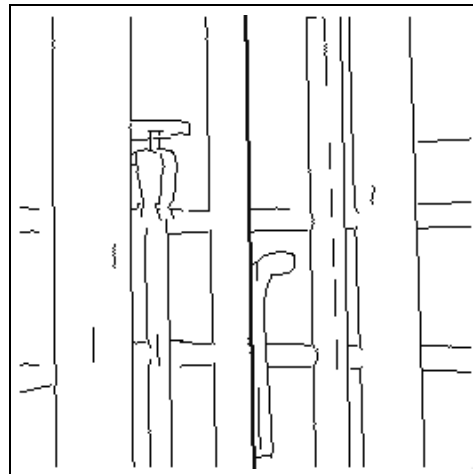


Figure 4.8 : Droite support du segment sélectionné dans la première image.

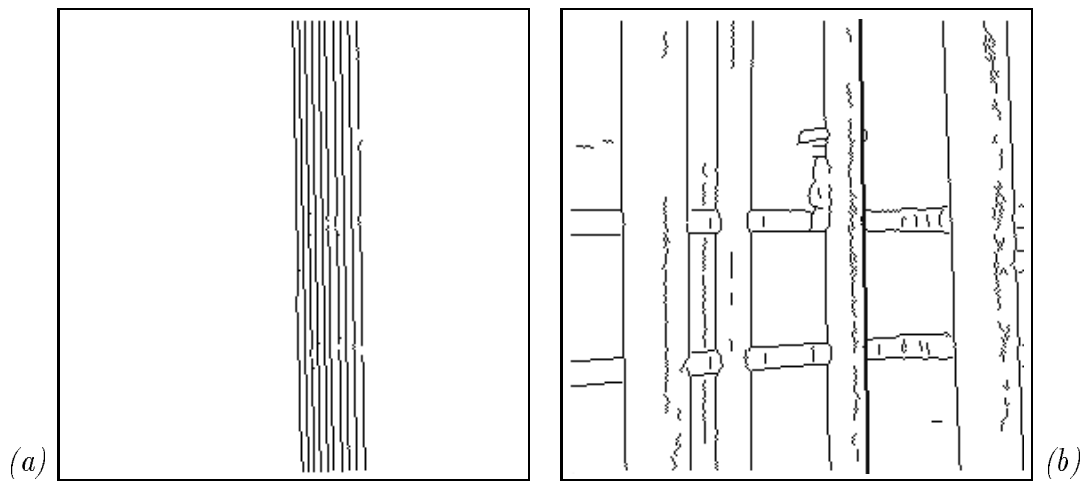


Figure 4.9 : (a) Séquence de segments correspondant au segment sélectionné. (b) Position finale du segment estimé en accord avec le segment extrait de cette image.

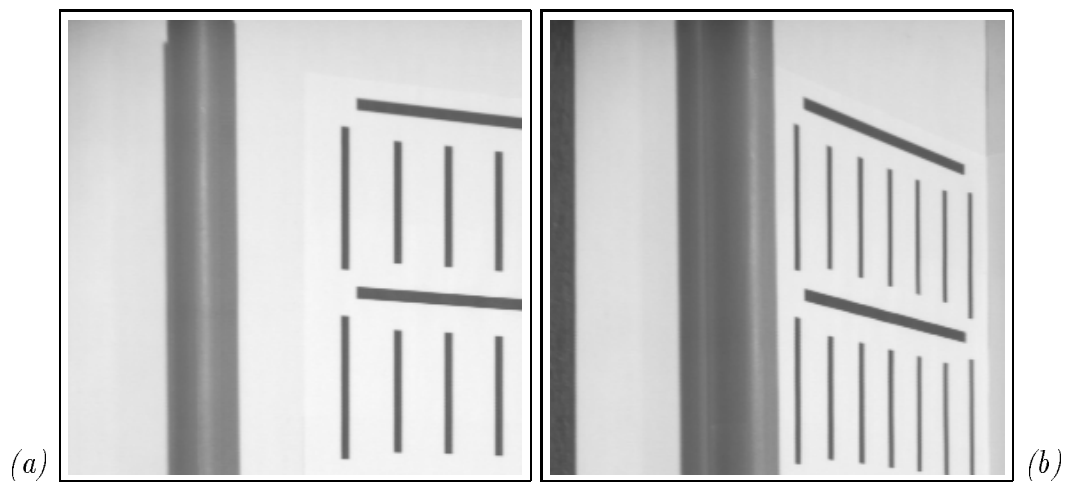


Figure 4.10 : (a) Première image de la séquence “mire”; taille : 256×256 . (b) Dernière image de la séquence “mire”. (Provenance : CEA-LETI-DEIN Saclay).

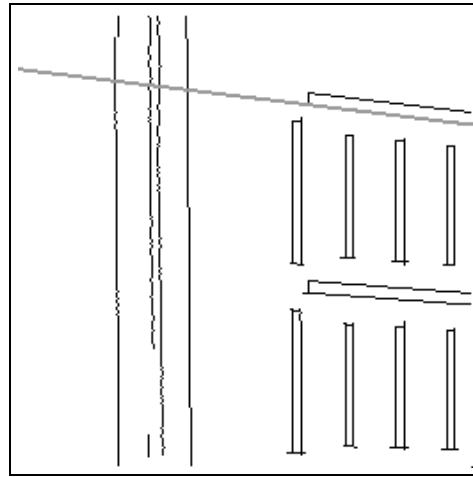


Figure 4.11 : Droite support du segment sélectionné dans la première image.

cellents résultats en dépit du mouvement complexe de la caméra (voir figure 4.12). Notons qu'une approche globale aurait eu des difficultés pour faire face à la très forte ambiguïté présente dans cette séquence. En effet, deux segments très proches ayant la même taille et la même orientation sont candidats pour l'appariement. On peut donc relever la robustesse de notre algorithme qui a pu distinguer le bon segment de celui situé juste au dessus (voir la figure (4.12)).

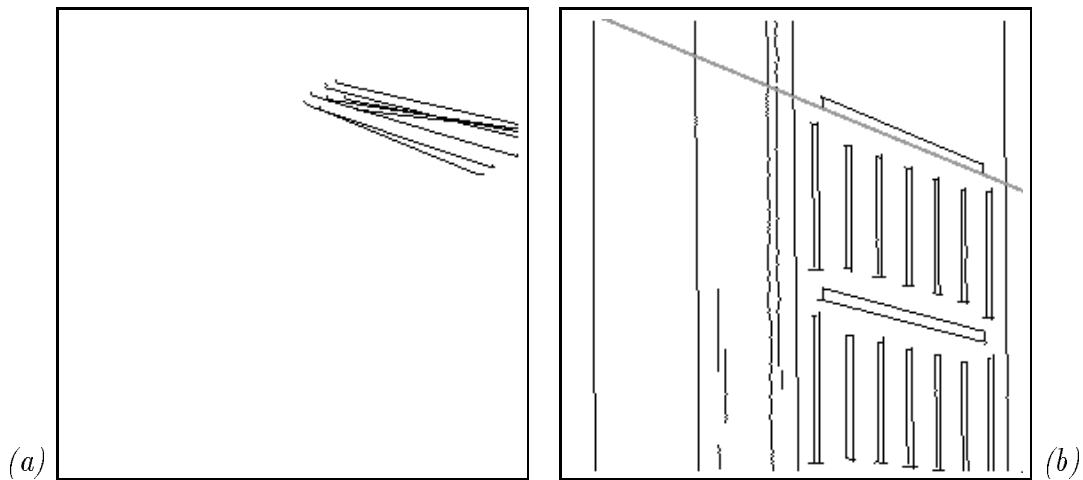


Figure 4.12 : (a) Séquence de segments correspondant au segment sélectionné. (b) Position finale du segment estimé en accord avec le segment extrait de cette image.

4.4.4 Implémentation temps-réel

Nos algorithmes ont été portés sur la machine Sympati2 [BAS88] qui est un système multiprocesseur parfaitement adapté aux algorithmes de traitement d'images. Ce système a été classé au premier rang lors du "Abingdon cross benchmark" [PRE89]. Cette machine a permis d'atteindre des performances de calcul remarquables sur nos modules de focalisation et de suivi.

1) Etape de focalisation : elle est relativement gourmande en temps de calcul et dépend de la densité des contours contenus dans la première image de la séquence. Par exemple, le temps d'exécution de cette étape est de 232 ms pour la séquence "table" dont 40 ms pour la détection des contours spatiaux (Shen-Castan), 184 ms pour la transformée de Hough (connue pour son coût calcul) et 8 ms pour le chaînage. Pour la séquence "tuyaux", ce temps est de 303 ms dont 40 ms pour la détection des contours spatiaux, 253 ms pour la transformée de Hough et 10 ms pour le chaînage. Ce temps est, comme on pouvait facilement le prévoir, plus important que dans le cas de la séquence "table" puisque la densité des contours spatiaux de la première image de la séquence est ici plus importante.

Notons que la lourdeur relative de cette étape ne compromet en aucun cas les performances de notre algorithme de reconstruction 3D en boucle fermée, vu qu'elle ne s'effectue qu'une seule fois au début du suivi d'un segment. Par contre, il est nécessaire que l'étape de suivi du segment qui, elle, est répétitive, soit proche du temps réel. Il ne s'agit pas nécessairement de la cadence vidéo mais plutôt de la fréquence d'échantillonnage de la boucle fermée pour une réalisation adéquate de la commande, soit une cadence minimale de l'ordre de 10 HZ.

1) Etape de suivi : le temps de calcul relatif à une itération de suivi du segment est inférieur à 35 ms. Ce temps dépend de la longueur du segment, de l'amplitude des déplacements et aussi de l'orientation du segment par rapport au segment de processeurs élémentaires (PEs) qui peut être vertical ou horizontal (voir l'annexe C consacré à la description du système Sympati2). Ce temps est en moyenne de 28 ms pour la séquence "mire", de 30 ms pour la séquence "table" et de 32 ms pour la séquence "tuyaux". Remarquons dans ce dernier cas que le temps de calcul reste du même ordre de grandeur que pour les autres séquences malgré la différence très significative de taille des segments considérés. Cela s'explique par le fait que cette fois on se trouve dans un cas de figure très favorable où on exploite au maximum les 32 PEs disponibles puisque le segment à suivre est vertical. Il est important de

souligner ici que notre approche de reconstruction 3D optimale de cylindres consiste justement à se ramener à des configurations particulières où le cylindre apparaît dans l'image sous la forme de deux droites verticales ou horizontales symétriques par rapport au centre de l'image [BOU92b], [CHA92], [BOU93c]. Cela permet donc d'assurer outre l'optimisation de l'estimation de la structure spatiale 3D du cylindre, des performances de calcul tout à fait remarquables sur Sympati2.

Rappelons que l'amplitude maximale des déplacements recherchés est ici de ± 7 . Or, la zone de recherche sera de plus en plus restreinte au cours de la convergence de la loi de commande. On pourra alors se limiter à une amplitude de ± 2 par exemple qui divise ce temps pratiquement par deux.

On obtient donc de très bonnes performances (bien en dessous des 100 ms requises) tout à fait compatibles avec les exigences de notre approche de reconstruction 3D par vision active.

- **Remarque :** Cet algorithme d'appariement de segments a été aussi porté sur la carte EDIXIA-1000 dont dispose l'IRISA. Les performances obtenues sont de l'ordre de 1 s pour la phase de focalisation et de 20 à 60 ms pour la phase de suivi. Dans la version implantée, on ne considère qu'un sous-échantillonnage des éléments de la chaîne de points contour (une quarantaine de points environ alors que la version effectuée par Sympati2 traite la chaîne complète) afin que cette carte mono-processeur puisse fournir des résultats satisfaisants. Notons que le suivi simultané de deux segments a été aussi implanté sur la carte EDIXIA-1000. Les performances obtenues (de l'ordre de 60 à 80 ms) restent très intéressantes. Des illustrations de ce suivi simultané de deux segments seront présentés dans le chapitre suivant consacré notamment à la reconstruction de primitives cylindriques.

4.5 Conclusion

Nous avons présenté dans ce chapitre une approche locale pour la mise en correspondance de segments dans une séquence d'images, opération indispensable pour la reconstruction 3D de cylindres qui est l'objectif final de cette étude.

La méthode développée devait faire face à des contraintes quasi temps réel au niveau de la phase d'analyse d'images. Il fallait donc, outre les aspects de robustesse et de fiabilité, prendre en compte l'aspect temps réel dans le choix d'une méthodologie appropriée pour l'appariement des limbes des cylindres dans la séquence d'images.

Nous avons conçu une méthode originale parfaitement adaptée aux besoins de notre application tant en ce qui concerne les implications algorithmiques que les aspects de robustesse. Cette méthode comprend un noyau basé sur une modélisation spatio-temporelle d'un élément de contour en mouvement (ECM) dont une version améliorée a été conçue. Un test robuste de vraisemblance permet de détecter et de suivre les ECM. La mise en œuvre effective de ce noyau est équivalente à un simple détecteur de contour spatial de type gradient-convolution. Les autres étapes de la méthode ne demandent que très peu de calcul. Une machine parallèle particulièrement adaptée à ce type d'opérations est la machine Sympati2, une carte multi-processeurs de type SIMD, adaptée au traitement d'images. Notre algorithme de suivi de segments a donc été porté sur cette machine ainsi que sur une carte monoprocesseur, et a atteint des performances de calcul qui répondaient aux exigences de notre approche de reconstruction 3D en boucle fermée. Outre ses performances temps réel, cet algorithme s'est avéré très efficace lors de sa validation sur plusieurs séquences d'images.

Notre technique, dont les motivations initiales étaient des applications en environnement nucléaire, est évidemment utilisable pour d'autres types de scènes. La méthode peut également être assez directement adaptée à d'autres primitives que des segments de droite. En effet, puisqu'elle ne manipule explicitement que des chaînes de contour, d'autres courbes paramétrées comme des arcs de cercle ou d'ellipses peuvent être considérées. Un aspect novateur de cette approche est qu'elle formalise et exploite réellement et efficacement le fait que toute problématique vision n'est pas une fin en soi mais est couplée à une action ou à une décision.

Chapitre 5

Résultats expérimentaux

Dans ce dernier chapitre, nous présentons les résultats qui ont été obtenus sur un banc expérimental et qui portent sur la reconstruction de trois primitives géométriques à savoir, le point, la droite et le cylindre. Ces résultats sont issus des méthodes de reconstruction 3D ainsi que des algorithmes de traitement d'images présentés dans les chapitres précédents.

5.1 Description du banc expérimental

Le banc expérimental dont nous disposons à l'IRISA est composé d'une caméra CCD PULNIX embarquée sur un robot cartésien AFMA à 6 degrés de liberté (voir figure 5.1).

La calibration de la caméra ainsi que l'identification de la matrice de passage entre le poignet du robot et le repère de la caméra ont été effectués [CHA89]. Ces procédures sont indispensables pour d'une part, transmettre correctement les consignes calculées par la commande au robot et, d'autre part, pour assurer la cohérence entre les données 2D mesurées dans l'image avec les données 3D estimées (passage repère de balayage \rightarrow repère caméra (voir annexe B), passage pixel \rightarrow mm).

Le torseur cinématique T_c , calculé dans le repère caméra, peut ainsi être exprimé dans l'espace articulaire en utilisant le jacobien inverse $J^{-1}(\underline{q})$ du robot :

$$\underline{\dot{q}}_c = J^{-1}(\underline{q})T_c \quad (5.1)$$



Figure 5.1 : *Banc expérimental*

La carte de traitement d'images utilisée est la carte EDIXIA 1000. Celle-ci permet l'acquisition d'images à la cadence vidéo (20 ms pour l'acquisition d'une trame). Les dimensions de la trame mémorisée sont de 256 lignes et 730 colonnes. Comme évoqué au chapitre précédent, l'algorithme de sélection d'un segment de la scène ainsi que l'algorithme de suivi de segments que nous avons développés ont été portés sur cette carte mono-processeur.

Sur cette carte, la phase de sélection consiste, une fois l'image acquise, à extraire d'abord à l'aide d'un opérateur Laplacien les contours de cette image. Ensuite, une fenêtre est positionnée sur la région d'intérêt de l'image de contours résultante et une transformée de Hough permet d'y sélectionner la droite ayant reçu le maximum de votes. Enfin, les points appartenant à la chaîne la plus longue sur cette droite support sont déterminés à l'aide d'une procédure de chaînage. Lorsque la fenêtre de recherche couvre entièrement l'image (taille 256×730), cette phase prend environ 1 s de temps de calcul.

Quant à l'algorithme de suivi qui va permettre d'apparier d'image à image les points du segment sélectionné, son implémentation sur cette carte a permis d'atteindre des performances quasi temps réel, soit 40 ms en moyenne par trame. Mais, pour des raisons de précision quant aux mesures du déplacement du robot, nous

considérons, dans toutes nos expérimentations, une période d'échantillonnage plus importante (80 ms) qui reste bien évidemment compatible avec les exigences de notre approche de reconstruction 3D en boucle fermée. Rappelons que sur cette carte, tous les points du segment ne sont pas appariés, seul un sous-ensemble est considéré afin d'accélérer le processus de suivi.

Par ailleurs, pour que la méthode de reconstruction 3D du cylindre basée sur deux limbes présentée au paragraphe 2.3.5 soit exploitable, ce même algorithme a été adapté au suivi simultané de deux segments ayant des orientations proches (le même filtre directionnel peut alors être utilisé). Les performances obtenues restent très intéressantes. Les temps de calcul sont de l'ordre de 60 ms par trame. Cela nous permettra de suivre simultanément les deux limbes d'un cylindre tout en respectant les contraintes de temps imposées par l'utilisation d'une commande en boucle fermée.

Nous allons à présent montrer les résultats obtenus pour la reconstruction de deux scènes différentes. La première scène, extrêmement simple, contient une cible bien visible sur laquelle nous nous focaliserons afin de reconstruire une primitive élémentaire : le point. La seconde scène, beaucoup plus complexe, illustre un environnement similaire à l'intérieur d'une centrale nucléaire. Nous allons ainsi montrer les résultats de reconstruction obtenus sur deux primitives très présentes dans un environnement nucléaire à savoir, le cylindre (ou tuyau) et la droite (ou arête). Dans le cas du cylindre, nous présenterons d'abord les résultats obtenus en utilisant un seul limbe, puis ceux, nettement meilleurs, obtenus en utilisant les deux limbes (voir les résultats de simulation aux paragraphes 2.3.6.3 et 3.4.4.3).

Dans ce qui suit, tous les paramètres 3D estimés seront exprimés dans un repère fixe : le repère de la position initiale de la caméra. La connaissance du déplacement de la caméra entre chaque prise de vue permet de déterminer la transformation entre ce repère et les autres positions de la caméra (voir annexe A).

5.2 Reconstruction 3D d'un point

La figure 5.2 illustre la position initiale de la caméra par rapport à la cible d'intérêt qui est ici une balle blanche posée sur un tourne disque noir. En partant de cette position, nous allons reconstruire la position 3D de cette balle considérée comme un point. Nous allons d'abord présenter les résultats de reconstruction 3D de ce point en utilisant un mouvement quelconque de la caméra. Ensuite, nous présenterons comment notre stratégie de reconstruction 3D permet d'améliorer de manière très

significative l'estimation obtenue.

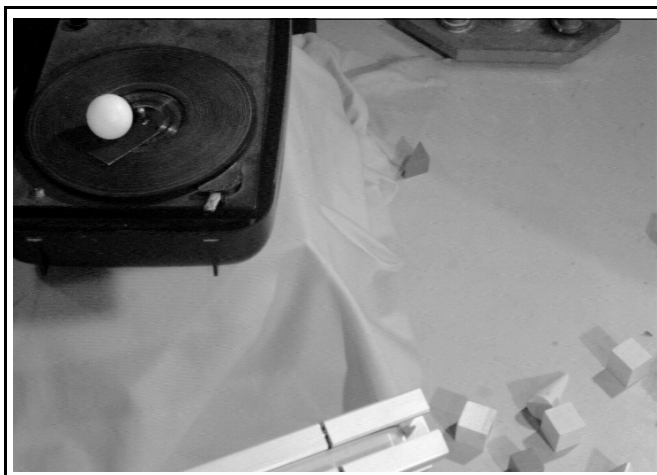


Figure 5.2 : *Position initiale entre la caméra et la cible de la scène*

5.2.1 Reconstruction 3D d'un point par vision dynamique

A partir de la position initiale de la figure 5.2, un mouvement quelconque est appliqué à la caméra (en l'occurrence $V_x = V_y = V_z = 50 \text{ mm/s}$, $\Omega_x = \Omega_y = \Omega_z = 3 \text{ dg/s}$) et, à chaque nouvelle acquisition d'image, on mesure les coordonnées (X, Y) du centre de gravité de l'ellipse observée dans l'image, leur déplacement ΔP par rapport à leur valeur précédente ainsi que le déplacement de la caméra. On estime alors, à chaque itération, les coordonnées (x_0, y_0, z_0) exprimées en mm du point 3D correspondant au point 2D observé. Les résultats obtenus sont illustrés sur la figure 5.3. Un simple filtre moyenneur (sur 3 valeurs) a été utilisé ici pour faire face aux éventuelles valeurs aberrantes. Ce même filtre a été utilisé dans toutes les expérimentations présentées dans ce chapitre.

La figure 5.4 montre l'évolution des coordonnées (X, Y) (exprimées en mm) du point 2D au cours du mouvement de la caméra. L'allure cyclique de ces courbes traduit le fait que le mouvement du robot n'est pas continu. En effet, on a été contraint, compte tenu des butées du robot, d'effectuer des mouvements d'allers-retours.

Sur les courbes de la figure 5.3, nous pouvons observer que les résultats obtenus sont très bruités. A l'erreur de discrétisation dont nous avons déjà pu constater les

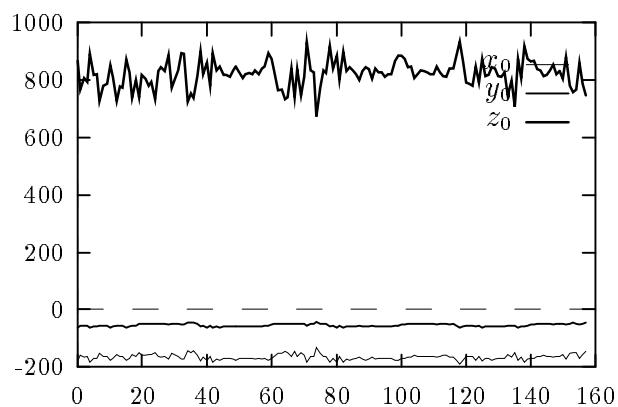


Figure 5.3 : *Reconstruction 3D du point sélectionné par vision dynamique : valeurs estimées à chaque itération des coordonnées 3D du point (en mm).*

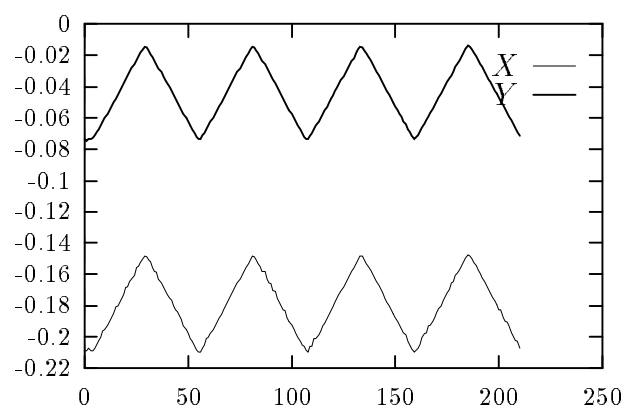


Figure 5.4 : *Paramètres 2D du point sélectionné sans stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées de la projection du point dans l'image.*

effets sur les résultats de simulation présentés au paragraphe 2.3.6, s'ajoutent ici les erreurs dues aux bruits de mesure.

5.2.2 Reconstruction 3D d'un point par vision active

5.2.2.1 Suppression de l'erreur de discrétisation

Comme nous l'avons montré au chapitre 3, les erreurs d'estimation observées dans le paragraphe précédent sont dues en partie au phénomène de discrétisation. Pour le contourner, nous avons montré qu'il suffisait de maintenir l'image et la surface de la primitive constantes tout au long de la trajectoire de la caméra ($\dot{P} = 0$ et $\dot{p}_0 = 0$). Cette stratégie ayant fait ses preuves en simulation (voir paragraphe 3.4.2), nous désirons l'appliquer ici au cas du point.

En partant de la même position initiale (voir figure 5.2), la commande référencée vision présentée au paragraphe 3.4.6, permet de générer de manière automatique les mouvements de la caméra qui assurent le maintien de la position du point fixe dans l'image.

La figure 5.5, structurée comme la figure 5.3, montre les résultats de reconstruction 3D obtenus en utilisant cette stratégie. En comparant ces résultats avec ceux de la figure 5.3, on peut noter l'amélioration très significative apportée par notre approche en boucle fermée par rapport au cas où aucune stratégie optimale sur le mouvement de la caméra n'est utilisée.

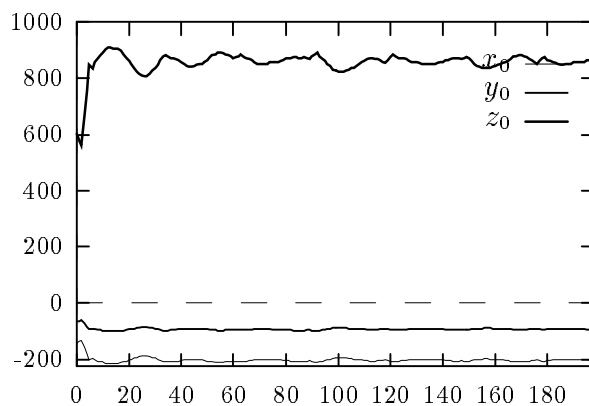


Figure 5.5 : *Reconstruction du point sélectionné en supprimant l'erreur de discrétisation (faible déplacement) : valeurs estimées à chaque itération des coordonnées 3D du point (en mm).*

Par ailleurs, pour atténuer les effets des erreurs de mesure du déplacement du robot, on peut considérer dans l'algorithme de reconstruction 3D non plus deux images successives mais plutôt deux images séparées par un segment de trajectoire de la caméra suffisamment important (10 fois plus important ici que dans le cas précédent). Cela est possible, malgré l'aspect continu de notre approche, car le point, grâce à l'asservissement visuel, reste fixe dans l'image et les contraintes $\dot{P} = 0$ et $\dot{p}_0 = 0$ sont toujours respectées. Les résultats obtenus sont de bien meilleure qualité que les précédents comme en témoignent les courbes de la figure 5.6.

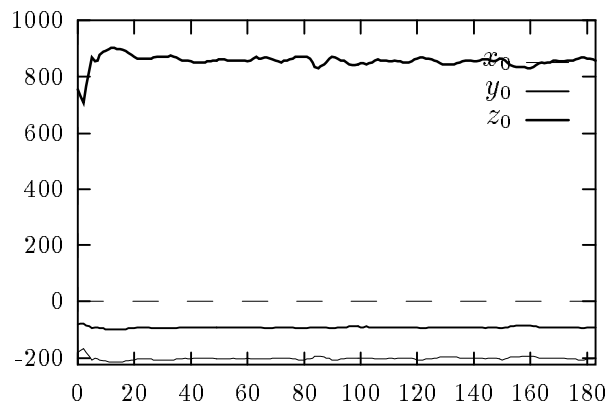


Figure 5.6 : *Reconstruction du point sélectionné en supprimant l'erreur de discrétisation (déplacement important) : valeurs estimées à chaque itération des coordonnées 3D du point (en mm).*

Sur le tableau 5.1 on peut voir les moyennes et les écarts type sur la position 3D estimée du point considéré évalués à partir de la vingtième itération.

	Moyenne	Ecart type
x_0	-202.5 mm	2.6 mm
y_0	-92.2 mm	1.8 mm
z_0	855.8 mm	8.3 mm

Tableau 5.1 : *Statistiques sur la position 3D estimée du point sélectionné en supprimant l'erreur de discrétisation.*

Nous allons voir à présent que l'on peut obtenir des estimations de bien meilleure qualité en utilisant notre stratégie optimale de reconstruction 3D.

5.2.2.2 Minimisation des effets des erreurs de mesure

Nous avons déjà vu au chapitre 3 que la reconstruction d'un point s'avère plus robuste quand la caméra est positionnée de telle manière que le point considéré apparaisse centré dans l'image. Toujours en partant de la même position initiale (voir figure 5.2), la position et l'orientation de la caméra sont contrôlées, en utilisant notre loi de commande en boucle fermée, afin d'atteindre la configuration optimale du point dans l'image illustrée par la figure 5.7.



Figure 5.7 : Position finale entre la caméra et la cible de la scène

Une fois le motif désiré atteint dans l'image, le mouvement optimal de la caméra est tel que la projection du point apparaisse en permanence fixe centré dans l'image ($\dot{P} = 0$) tandis que la caméra tourne autour de celui-ci. Notons que cette consigne, décrite au paragraphe 3.4.6.1, assure également la contrainte $\dot{p}_0 = 0$.

La figure 5.8 montre la convergence des paramètres 2D (X, Y) à la valeur finale désirée ($X^* = Y^* = 0$). On peut noter les quelques erreurs commises sur la fonction de tâche (aux environs de la 300ème itération) qui correspondent au moment où, la convergence ayant été établie, le mouvement secondaire propre au point est activé. Il est clair que sans cette tâche secondaire, la reconstruction 3D du point est impossible après convergence (mouvement nul lorsque $\underline{e} = 0$) comme nous l'avons déjà souligné au chapitre 3. Ces erreurs sont dues au fait que ce mouvement, calculé à partir de l'estimation courante des paramètres 3D, n'est pas en mesure d'assurer la contrainte $\dot{P} = 0$ en raison des estimations très mauvaises qui sont obtenues aux premières itérations (voir figure 5.9). Du point de vue de la commande, cela signifie que la tâche référencée-vision et la tâche secondaire ne sont pas compatibles, une erreur

d'estimation étant faite sur les opérateurs de projection (W^+ et $\mathbb{I}_6 - W^+W$).

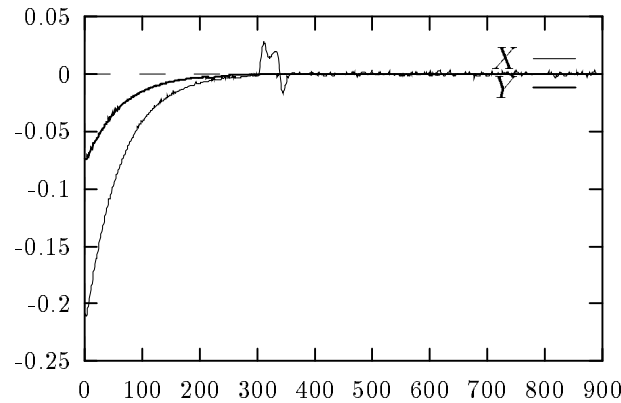


Figure 5.8 : Paramètres 2D du point sélectionné avec une stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées de la projection du point dans l'image.

Après quelques itérations de la boucle de commande, ces erreurs sont très rapidement atténuées et les contraintes $\dot{P} = 0$ et $\dot{p}_0 = 0$ sont de nouveau respectées, ce qui explique l'amélioration très importante observée sur l'estimation des paramètres 3D du point.

La figure 5.9, structurée comme la figure 5.3, montre les résultats de reconstruction 3D obtenus après convergence en utilisant cette stratégie. Sur ces courbes, on peut constater que quelques itérations suffisent pour affiner et stabiliser les estimations sur les paramètres 3D. Comme en simulation (voir paragraphe 3.4.4.1), ces résultats s'avèrent de meilleure qualité que ceux obtenus en supprimant juste l'erreur de discrétisation (voir figure 5.5). En utilisant des déplacements plus importants entre les deux images prises en compte dans le processus de reconstruction du point considéré (du même ordre que ceux utilisés dans le paragraphe précédent), une amélioration très notable de la qualité de l'estimation est obtenue comme en témoignent les courbes de la figure 5.10 que l'on peut comparer une fois de plus aux courbes de la figure 5.6. Le tableau 5.2 qui représente les moyennes et les écarts type sur la position 3D estimée du point considéré (évalués à partir de la vingtième itération) confirme la stabilité et la robustesse de notre approche. Notons que ces statistiques s'avèrent également bien plus intéressantes que celles obtenus en supprimant juste l'erreur de discrétisation (voir tableau 5.1).

Ayant validé avec succès notre approche sur le point, nous allons à présent nous attacher à reconstruire des primitives plus complexes : les droites et les cylindres.

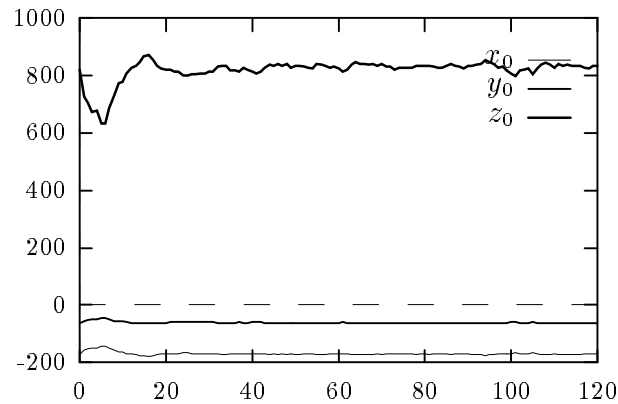


Figure 5.9 : *Reconstruction optimale du point sélectionné (faible déplacement) : valeurs estimées à chaque itération des coordonnées 3D du point (en mm).*

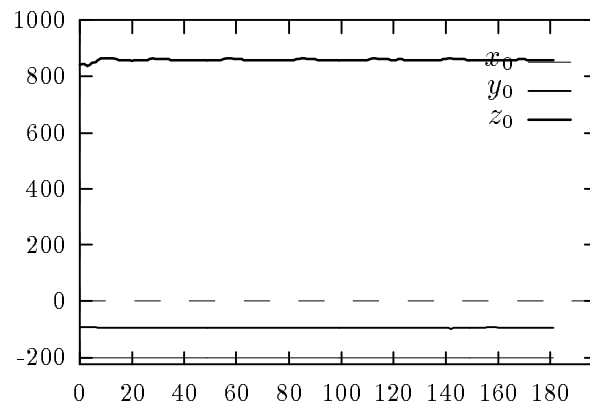


Figure 5.10 : *Reconstruction optimale du point sélectionné (déplacement important) : valeurs estimées à chaque itération des coordonnées 3D du point (en mm).*

	Moyenne	Ecart type
x_0	-201.8 mm	0.2 mm
y_0	-95.2 mm	0.6 mm
z_0	859.0 mm	1.6 mm

Tableau 5.2 : *Statistiques sur la position 3D estimée du point sélectionné en minimisant les effets des erreurs de mesure.*

Contrairement au cas précédent qui ne nécessitait qu'un traitement d'images élémentaire (extraction du centre de gravité de la cible dans l'image), ces deux primitives requièrent un traitement d'images plus élaboré : le suivi de segments 2D dans les images successives.

5.3 Reconstruction 3D d'une droite

La figure 5.11 montre l'image initiale d'une scène constituée de plusieurs tuyaux. Cette scène illustre ainsi parfaitement un environnement nucléaire. Sur cette image est représentée (en blanc) la droite support du segment qui a été sélectionnée initialement par la transformée de Hough. En partant de cette position, nous allons, comme pour le point, d'abord présenter les résultats de reconstruction 3D de cette droite en utilisant un mouvement quelconque de la caméra. Nous présenterons ensuite, les résultats, nettement plus robustes, obtenus en utilisant notre approche de reconstruction 3D par vision active.

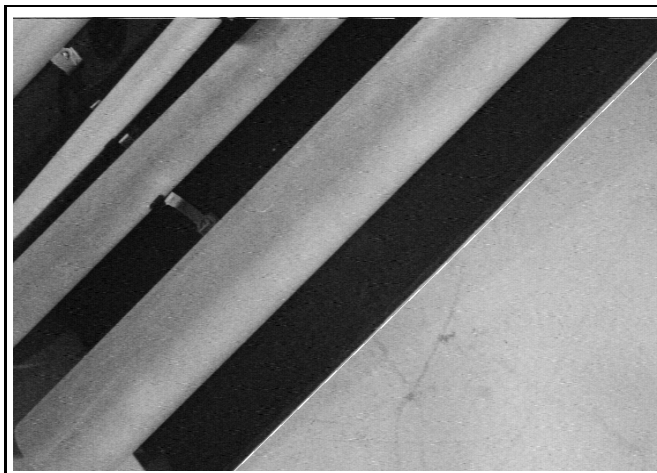


Figure 5.11 : *Position initiale entre la caméra et la droite sélectionnée*

5.3.1 Reconstruction 3D d'une droite par vision dynamique

A partir de la position initiale de la figure 5.11, un mouvement quelconque est appliqué à la caméra (le même que pour le point). Bien entendu, à chaque nouvelle

acquisition, l'algorithme de suivi de segments est activé pour déterminer les paramètres (ρ, θ) du segment correspondant à celui qui a été initialement sélectionné. On peut alors estimer les paramètres 3D définissant la droite correspondante de la scène en utilisant la méthode présentée au paragraphe 2.3.2.

La droite ayant été modélisée comme l'intersection de deux plans orthogonaux P_1 et P_2 , P_1 passant par le centre optique, 7 paramètres sont estimés : les paramètres a_1, b_1 et c_1 du plan P_1 et les paramètres a_2, b_2, c_2 et d_2 du plan P_2 . Les résultats d'estimation obtenus sont illustrés sur la figure 5.13 : en haut, à gauche, les différentes valeurs de a_1, b_1, c_1 ; à droite, les différentes valeurs de a_2, b_2, c_2 et en bas, les différentes valeurs de d_2 en *mm*. La figure 5.12 montre l'évolution des paramètres (ρ, θ) exprimées respectivement en *mm* et en *dg* (la valeur de ρ est multipliée par un facteur 100) de la droite au cours du mouvement de la caméra.

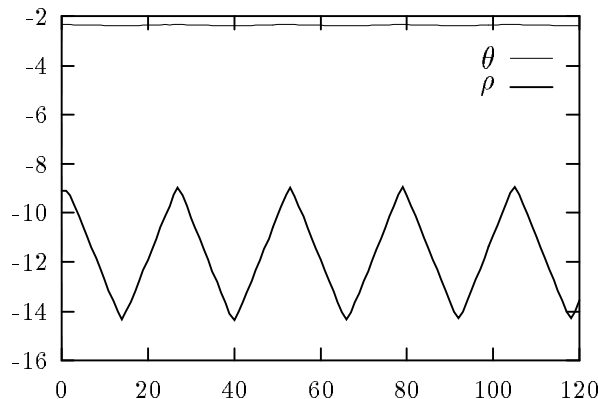


Figure 5.12 : Paramètres 2D de la droite sélectionnée sans stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées polaires de la projection de la droite dans l'image.

Sur ces courbes, nous pouvons observer que les résultats obtenus pour a_1, b_1 et c_1 sont assez bons. Ceci est logique puisque ces valeurs sont obtenues directement à partir de l'image d'une seule droite. Par contre, les autres valeurs qui sont déterminées à partir de deux images successives sont, comme dans le cas du point, très bruitées. Nous pouvons constater aussi à travers ces courbes que le bruit sur la mesure du déplacement du robot a relativement plus d'influence sur la qualité des estimations que le bruit provenant de la mesure des informations visuelles. En effet, les paramètres du plan P_1 , ne dépendant que des informations image, sont beaucoup moins bruités que ceux du plan P_2 qui, eux, sont directement liés au mouvement de

la caméra. Bien entendu, ce phénomène pourrait être aussi interprété comme le fait que le bruit sur la mesure du déplacement du robot est plus prononcé que celui sur les mesures image. Néanmoins, ce phénomène a été observé également en simulation et ce, en affectant les deux types de mesures par des bruits du même ordre.

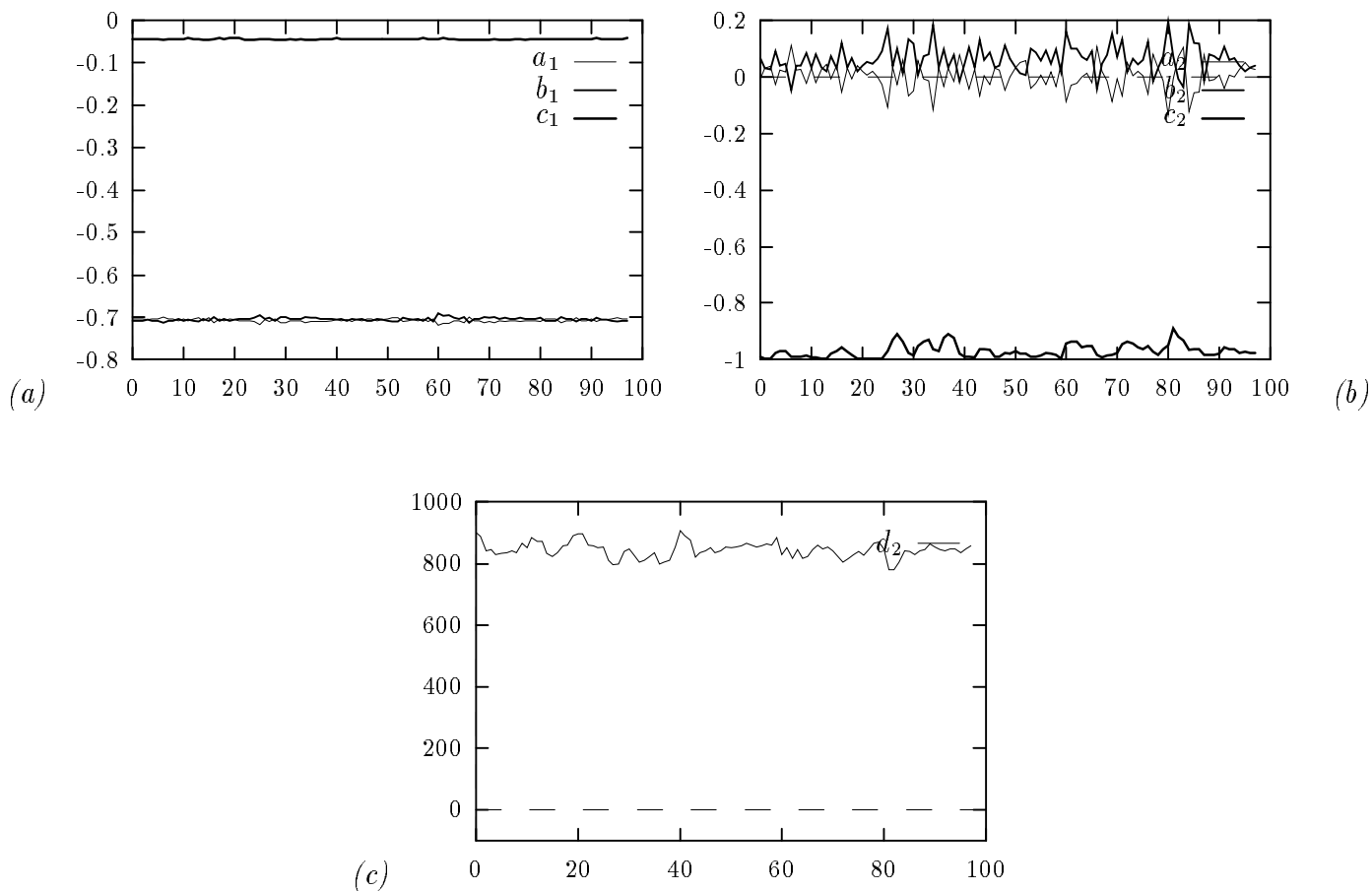


Figure 5.13 : *Reconstruction 3D de la droite sélectionnée par vision dynamique : (a) valeurs estimées à chaque itération des paramètres a_1, b_1 et c_1 définissant le plan, contenant la droite, passant par le centre optique. (b) et (c) valeurs estimées à chaque itération des paramètres a_2, b_2, c_2 et d_2 (en mm) du second plan contenant la droite.*

5.3.2 Reconstruction 3D d'une droite par vision active

La position optimale pour la reconstruction d'une droite est telle qu'elle apparaisse centrée verticale ou horizontale dans l'image comme nous l'avons vu au chapitre 3.

Comme dans le cas du point, l'asservissement visuel permet, en partant de la position initiale de la figure 5.11, d'atteindre une configuration optimale de la droite dans l'image (une droite verticale centrée ici). La figure 5.15 montre (en blanc) la configuration finale dans l'image de la droite considérée. Une fois l'asservissement complètement réalisé, le mouvement optimal de la caméra est tel que la projection de la droite apparaisse en permanence fixe centrée dans l'image ($\dot{P} = 0$) tandis que la caméra tourne autour de celle-ci. Notons que cette consigne, comme pour le point, assure également la contrainte $\dot{p}_0 = 0$ (voir le paragraphe 3.4.6.2).

La figure 5.14 montre la convergence des paramètres (ρ, θ) à la valeur finale désirée ($\rho^* = 0, \theta^* = \pi$). L'allure de ces courbes, similaire à celle des courbes représentant les paramètres (X, Y) du point, traduit encore une fois la robustesse et la bonne convergence de notre loi de commande en boucle fermée.

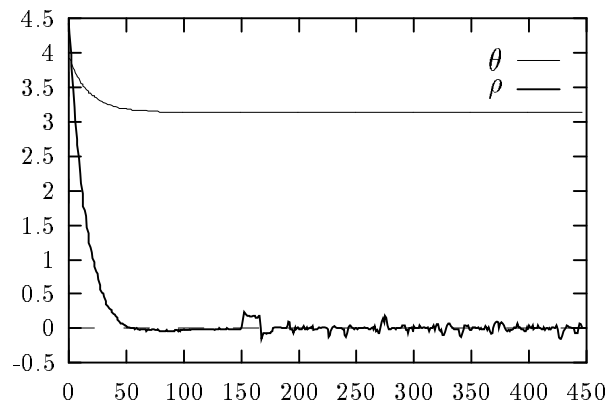


Figure 5.14 : Paramètres 2D de la droite sélectionnée avec une stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées polaires de la projection de la droite dans l'image.

La figure 5.16, structurée comme la figure 5.13, montre les résultats de reconstruction 3D obtenus après convergence en utilisant cette stratégie. On peut constater ici aussi que quelques itérations suffisent pour affiner et stabiliser les estimations sur les paramètres 3D. En comparant ces résultats avec ceux de la figure 5.13, on peut noter l'amélioration très importante apportée par notre approche en boucle fermée par rapport au cas où aucune stratégie optimale sur le mouvement de la caméra n'est utilisée. Le tableau 5.3 récapitule les moyennes et les écarts types sur les paramètres 3D estimés de la droite considérée (évalués comme dans le cas du point à partir de la vingtième itération).

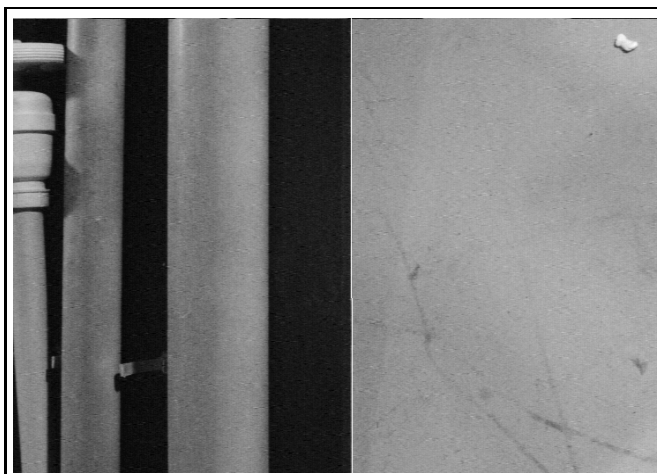


Figure 5.15 : *Position optimale entre la caméra et la droite sélectionnée*

Par ailleurs, pour améliorer la robustesse de ces estimations, nous considérons, comme dans le cas du point, de forts déplacements entre les deux images prises en compte dans le processus de reconstruction 3D de la droite considérée. Cela permet, comme on peut le noter sur la figure 5.17, d'obtenir des résultats nettement meilleurs que les précédents.

	Moyenne	Ecart type
a_1	-0.69	0.001
b_1	-0.72	0.001
c_1	-0.05	0.0006
a_2	0.013	0.003
b_2	0.054	0.004
c_2	-0.99	0.0003
d_2	856.6 mm	1.4 mm

Tableau 5.3 : *Statistiques sur les paramètres 3D estimés de la droite sélectionnée (où a_1, b_1 et c_1 définissent le plan passant par le centre optique et a_2, b_2, c_2 et d_2 le second plan caractérisant la droite)*

Il est important de souligner encore une fois la robustesse et les performances temps-réel de notre algorithme de suivi de segments qui se sont traduit ici par une qualité de reconstruction tout à fait satisfaisante. Bien entendu, des erreurs de reconstruction persistent encore qui proviennent notamment des imprécisions qui affectent les paramètres géométriques de la caméra issus d'un processus de calibrage, et des erreurs sur les mesures du déplacement du robot.

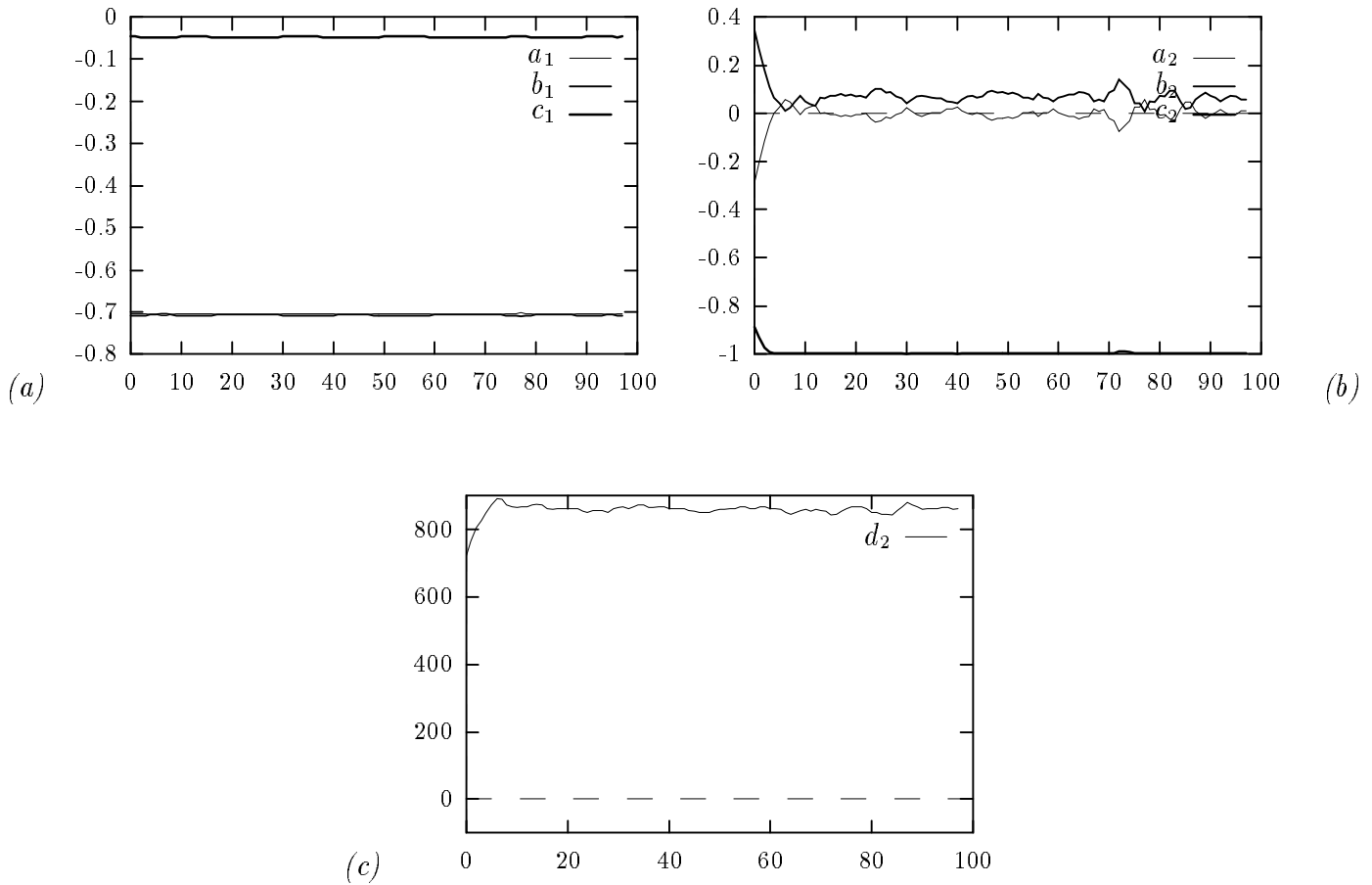


Figure 5.16 : *Reconstruction 3D de la droite sélectionnée par vision active (faible déplacement) : (a) valeurs estimées à chaque itération des paramètres a_1, b_1 et c_1 définissant le plan, contenant la droite, passant par le centre optique. (b) et (c) valeurs estimées à chaque itération des paramètres a_2, b_2, c_2 et d_2 (en mm) du second plan contenant la droite.*

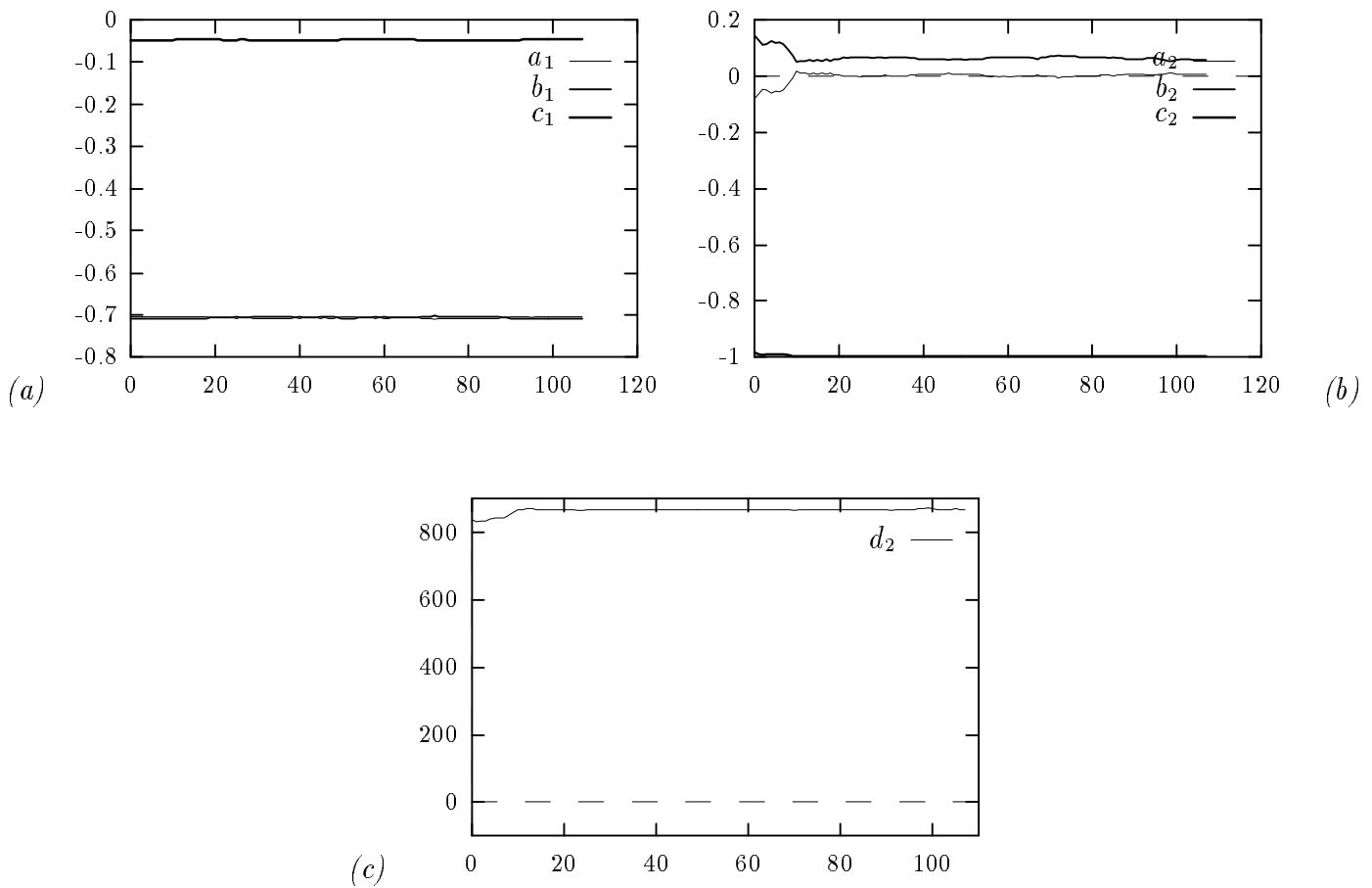


Figure 5.17 : *Reconstruction 3D de la droite sélectionnée par vision active (déplacement important) : (a) valeurs estimées à chaque itération des paramètres a_1, b_1 et c_1 définissant le plan, contenant la droite, passant par le centre optique. (b) et (c) valeurs estimées à chaque itération des paramètres a_2, b_2, c_2 et d_2 (en mm) du second plan contenant la droite.*

Nous allons maintenant présenter les résultats relatifs à la reconstruction d'une primitive tridimensionnelle : le cylindre.

5.4 Reconstruction 3D d'un cylindre

Dans ce paragraphe, nous allons reconstruire, toujours en utilisant la même scène expérimentale (voir figure 5.11), des primitives cylindriques. Deux exemples sont présentés ici : les résultats relatifs à la reconstruction du cylindre (de rayon 40 mm) situé tout à fait à droite de la scène et celui (de rayon 25 mm) situé juste à gauche de celui-ci. Nous les dénommerons respectivement *le premier cylindre* et *le second cylindre* de la scène.

Nous allons présenter tout d'abord les résultats de reconstruction issus de la méthode basée sur un seul limbe, présentée au paragraphe 2.3.5.3. Ces résultats, qui se sont avérés d'une qualité insuffisante notamment sur l'estimation du rayon du cylindre (rien de surprenant puisqu'un seul limbe du cylindre est observé), sont cependant indispensables pour identifier le second limbe du cylindre dans l'image. On peut alors faire appel à la méthode de reconstruction 3D, potentiellement plus robuste, basée, elle, sur l'utilisation des deux limbes du cylindre (voir paragraphe 2.3.5.2).

5.4.1 Méthode de reconstruction 3D basée sur un seul limbe

La figure 5.18 met en évidence (en blanc), la droite support initialement sélectionnée dans l'image et correspondant cette fois, non plus à une arête, mais à un des limbes du premier cylindre de la scène. Comme dans les deux cas précédents, nous allons d'abord présenter les résultats de reconstruction 3D obtenus en utilisant un mouvement quelconque (le même que dans les cas précédents) puis, ceux issus de notre approche par vision active.

5.4.1.1 Reconstruction d'un cylindre par vision dynamique

La figure 5.19, illustre les résultats obtenus sur l'estimation des paramètres 3D du premier cylindre de la scène en utilisant un mouvement quelconque et en effectuant, à chaque nouvelle acquisition d'image, le suivi du limbe initialement observé. En haut de cette figure, à gauche, sont présentées les différentes valeurs des paramètres

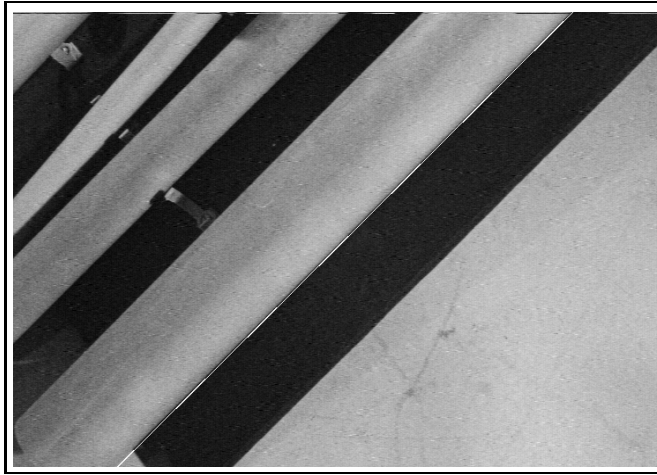


Figure 5.18 : *Position initiale entre la caméra et le limbe sélectionné du premier cylindre de la scène*

a , b et c du vecteur directeur de l'axe du cylindre, à droite les différentes valeurs des paramètres x_0, y_0, z_0 du point de son axe le plus proche du centre optique et de son rayon r , et, en bas, l'erreur entre le rayon estimé et sa valeur réelle (40 mm ici). Rappelons que la reconstruction 3D d'un cylindre en utilisant un seul de ses limbes nécessite au moins trois images du limbe considéré. Pour améliorer la robustesse de l'estimation, nous en utilisons quatre ici car nous effectuons, après l'acquisition de deux images successives, un déplacement important de la caméra avant de prendre en compte la seconde paire d'images.

Comme les deux cas précédents, l'utilisation d'un mouvement quelconque entraîne des erreurs d'estimation très importantes qui sont particulièrement catastrophiques sur le rayon en dépit du fort déplacement de la caméra qui a été effectuée.

La figure 5.20 montre les valeurs mesurées des paramètres (ρ, θ) de la droite support du limbe perçu au cours du déplacement de la caméra.

5.4.1.2 Reconstruction d'un cylindre par vision active

La figure 5.21 illustre la position finale de la caméra par rapport au limbe sélectionné du premier cylindre de la scène (une droite verticale centrée ici). Cette position optimale a été obtenue à partir de la position initiale de la figure 5.18 en contrôlant le mouvement de la caméra à l'aide de la loi de commande en boucle fermée. La figure 5.22 montre la convergence des paramètres (ρ, θ) à la valeur finale désirée ($\rho^* = 0, \theta^* = \pi$). Comme dans les deux cas précédents, on utilise, une fois la tâche

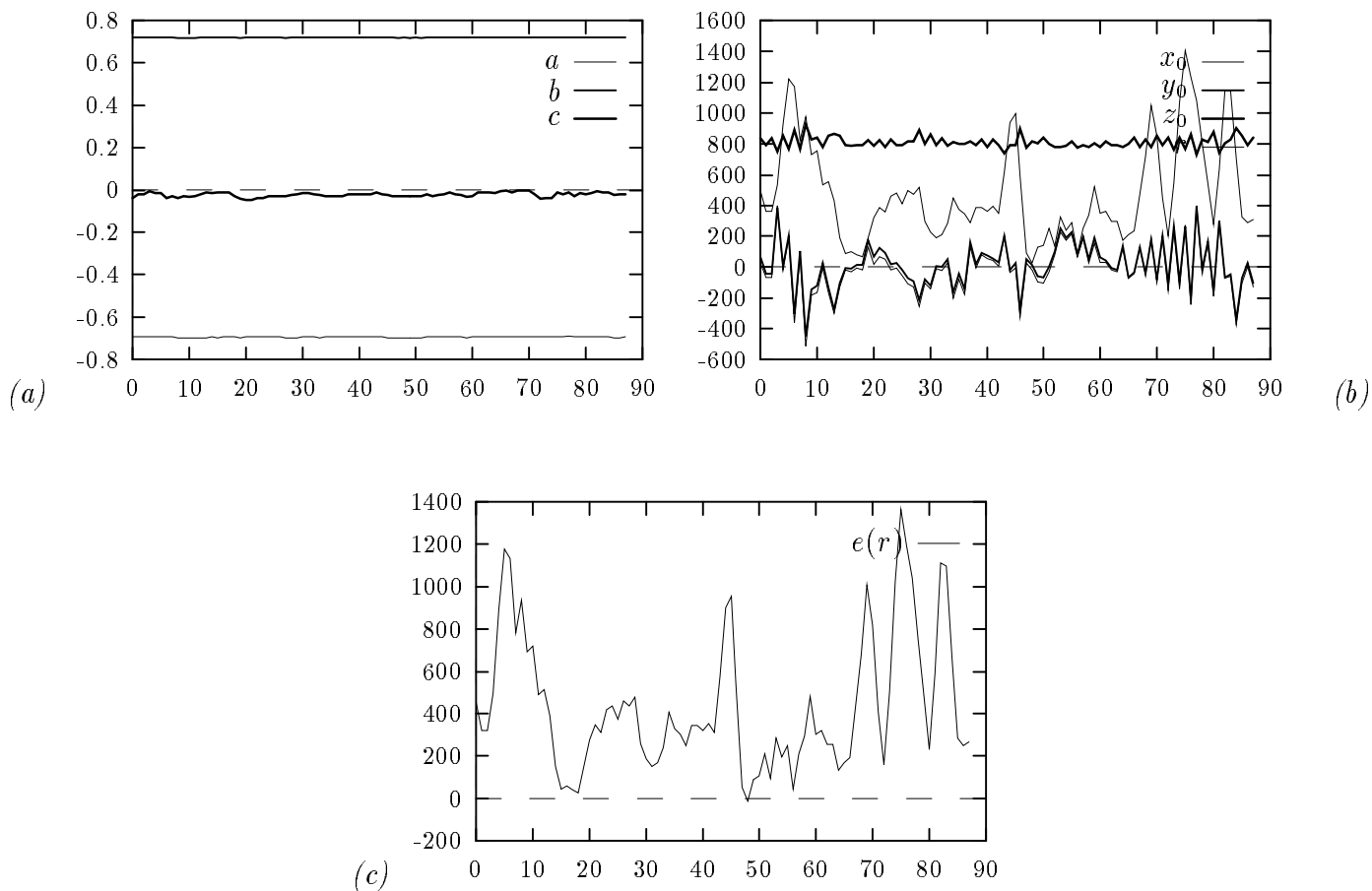


Figure 5.19 : *Reconstruction 3D du premier cylindre sélectionné de la scène par vision dynamique en utilisant la méthode basée sur un seul limbe (déplacement important) : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r (en mm). (c) erreur entre la valeur estimée du rayon et sa valeur réelle (40 mm).*

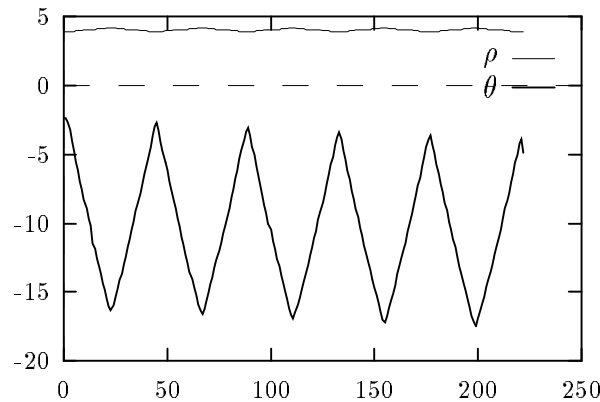


Figure 5.20 : Paramètres 2D du limbe sélectionné du premier cylindre de la scène sans stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées polaires de la droite correspondant à la projection du limbe du cylindre dans l'image.

de positionnement complètement réalisée ($\underline{\epsilon} = 0$), un mouvement secondaire, décrit au paragraphe 3.4.6.3, qui consiste à assurer simultanément les contraintes $\underline{\dot{P}} = 0$ et $\underline{\dot{p}_0} = 0$ tout en tournant autour de l'axe du cylindre.

La figure 5.23, configurée comme la figure 5.19, illustre les résultats de reconstruction 3D du cylindre considéré obtenus après convergence en utilisant cette stratégie optimale. Ces résultats ont été déterminés en utilisant trois images successives ce qui explique les importantes erreurs observées sur le rayon du cylindre. Il est en effet impossible d'acquérir une information fiable sur le rayon avec un faible déplacement de la caméra. En utilisant un déplacement important du même ordre que les précédents, on observe (voir figure 5.24) une très nette amélioration de la qualité de l'estimation notamment sur le rayon qui s'avère naturellement le plus sensible à cette procédure. Intuitivement, on pourrait imaginer qu'un mouvement couvrant la moitié du pourtour du cylindre (ce qui n'est pas le cas ici), compenserait complètement l'absence d'informations sur le second limbe de ce dernier. Autrement dit, plus le mouvement pris en compte est important et plus la robustesse de l'estimation de la structure 3D du cylindre est améliorée.

Les résultats obtenus, quoique d'une précision encore très insuffisante sur l'estimation du rayon, sont de loin beaucoup plus satisfaisants que ceux obtenus en utilisant un mouvement quelconque (voir figure 5.19). Ces différences de résultats, très significatives, mettent en valeur une fois de plus l'apport incontestable de la vision active par rapport à la vision dynamique sur l'estimation de la structure 3D des primitives de l'environnement.



Figure 5.21 : *Position optimale entre la caméra et le limbe sélectionné du premier cylindre de la scène*

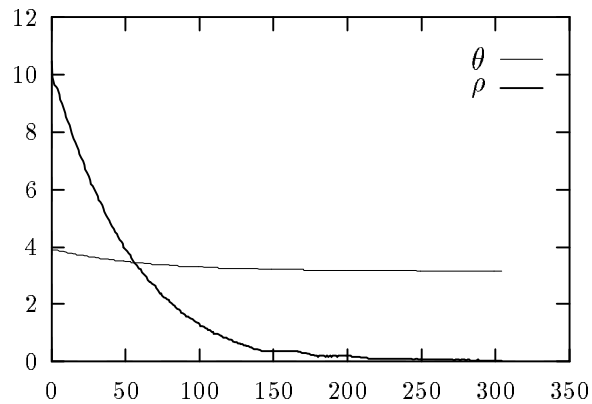


Figure 5.22 : *Paramètres 2D du limbe sélectionné du premier cylindre avec une stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées polaires de la droite correspondant à la projection du limbe du cylindre dans l'image.*

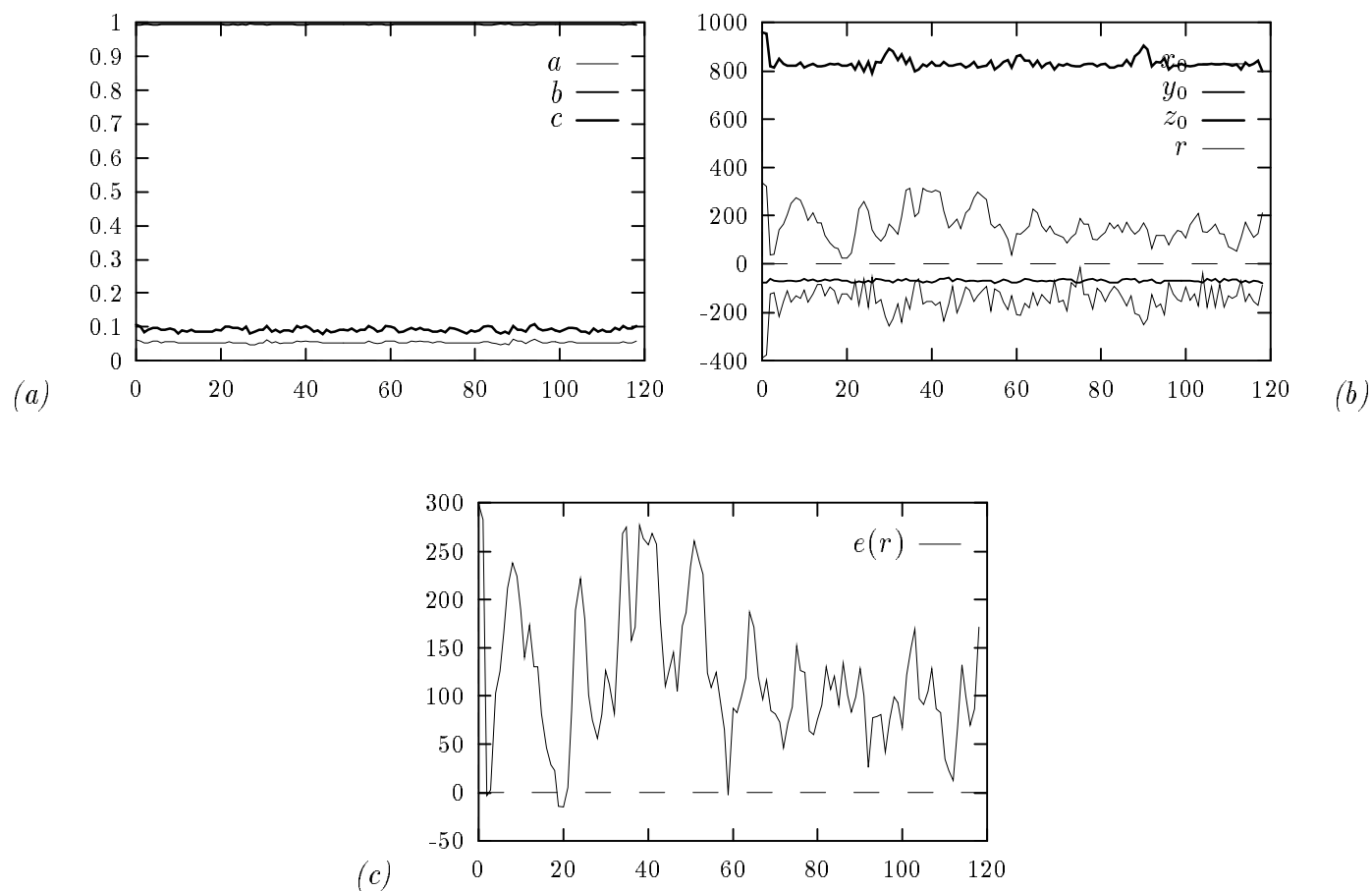


Figure 5.23 : *Reconstruction 3D du premier cylindre de la scène par vision active en utilisant la méthode basée sur un seul limbe (faible déplacement) : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r (en mm). (c) erreur entre la valeur estimée du rayon et sa valeur réelle (40 mm).*

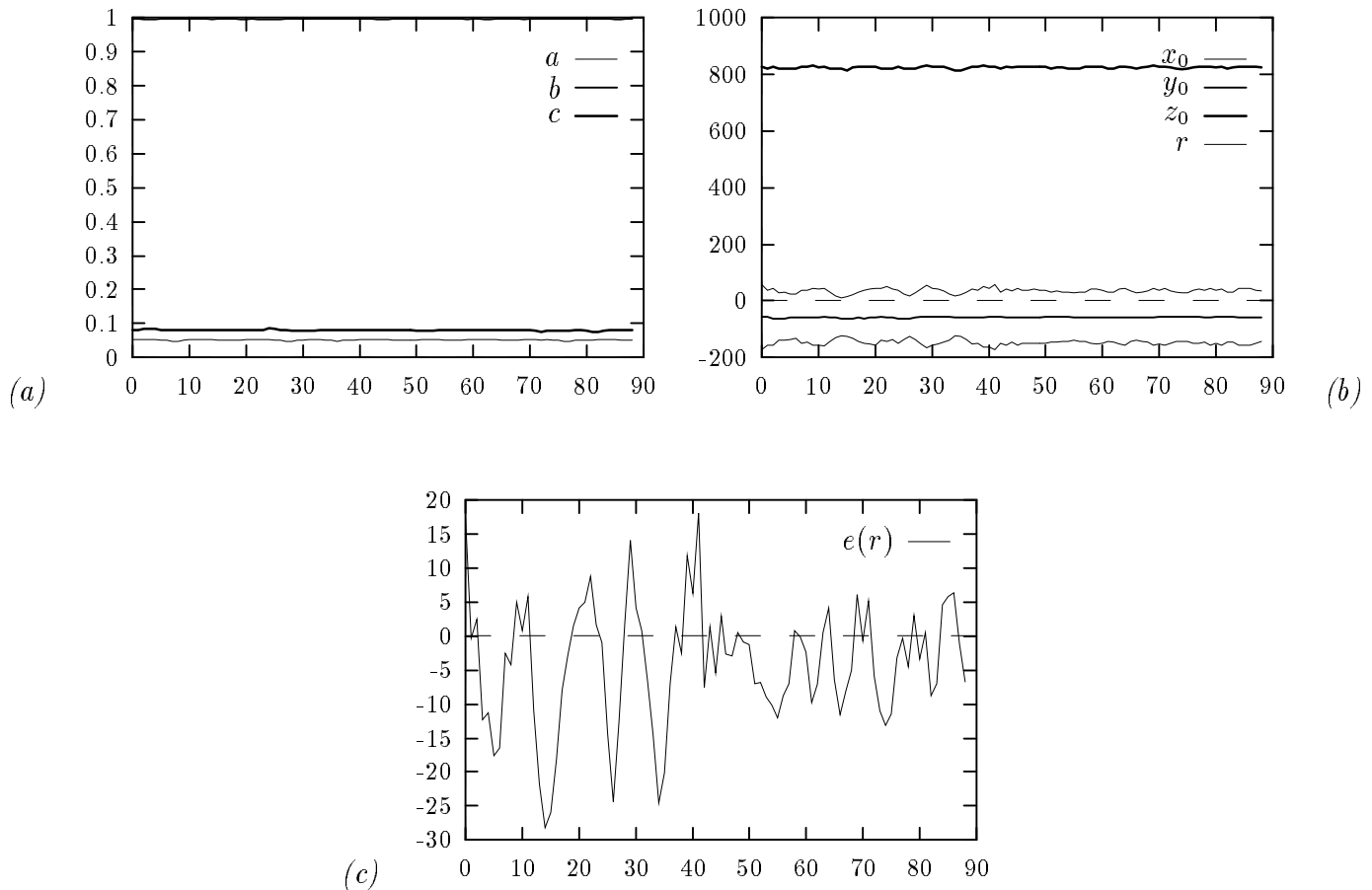


Figure 5.24 : Reconstruction 3D du premier cylindre de la scène par vision active en utilisant la méthode basée sur un seul limbe (déplacement important) : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r (en mm). (c) erreur entre la valeur estimée du rayon et sa valeur réelle (40 mm).

Toujours en utilisant la même méthode de reconstruction 3D de cylindres basée sur un seul limbe, on s'intéresse cette fois à l'estimation de la structure du second cylindre de la scène, situé juste à gauche de celui qui vient d'être reconstruit. Pour ce cylindre, nous présentons juste les résultats obtenus en utilisant notre approche en boucle fermée. En partant de la position initiale du limbe sélectionné de ce cylindre illustrée par la figure 5.25, une tâche de positionnement, semblable à celle utilisée dans le cas du cylindre précédent, est déclenchée. La figure 5.26 illustre la position finale atteinte entre la caméra et le limbe considéré après complète réalisation de cette tâche.

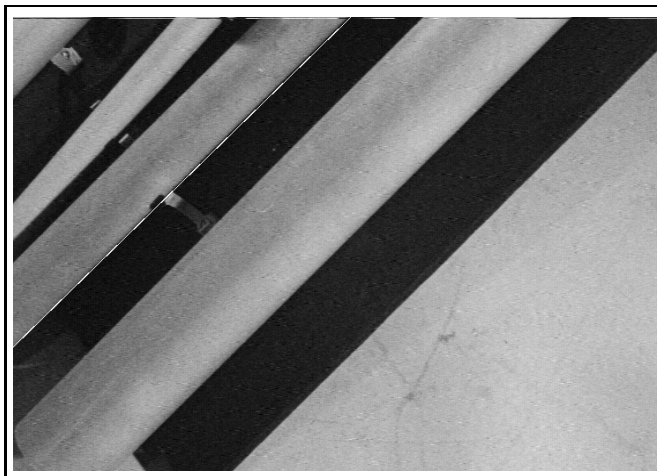


Figure 5.25 : *Position initiale entre la caméra et le limbe sélectionné du second cylindre de la scène*

Sur la figure 5.27, on peut noter, comme dans le cas du premier cylindre, la bonne convergence de la loi de commande en boucle fermée vers la configuration optimale du limbe désirée dans l'image.

Les résultats de reconstruction 3D obtenus sur ce cylindre sont montrés sur la figure 5.28, illustrée comme la figure 5.24. Les courbes obtenues ont une allure similaire à celles observées dans le cas du premier cylindre. La qualité de l'estimation du rayon s'avère également assez pauvre, ce qui nous incite à faire appel à la méthode, bien plus robuste, qui intègre la totalité des informations 2D perçues sur le cylindre à savoir, la projection de ses deux limbes dans l'image.



Figure 5.26 : *Position optimale entre la caméra et le limbe sélectionné du second cylindre de la scène*

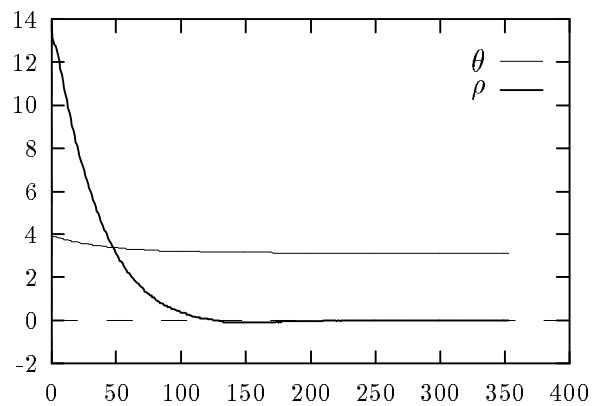


Figure 5.27 : *Paramètres 2D du limbe sélectionné du second cylindre de la scène avec une stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées polaires de la droite correspondant à la projection du limbe du cylindre dans l'image.*

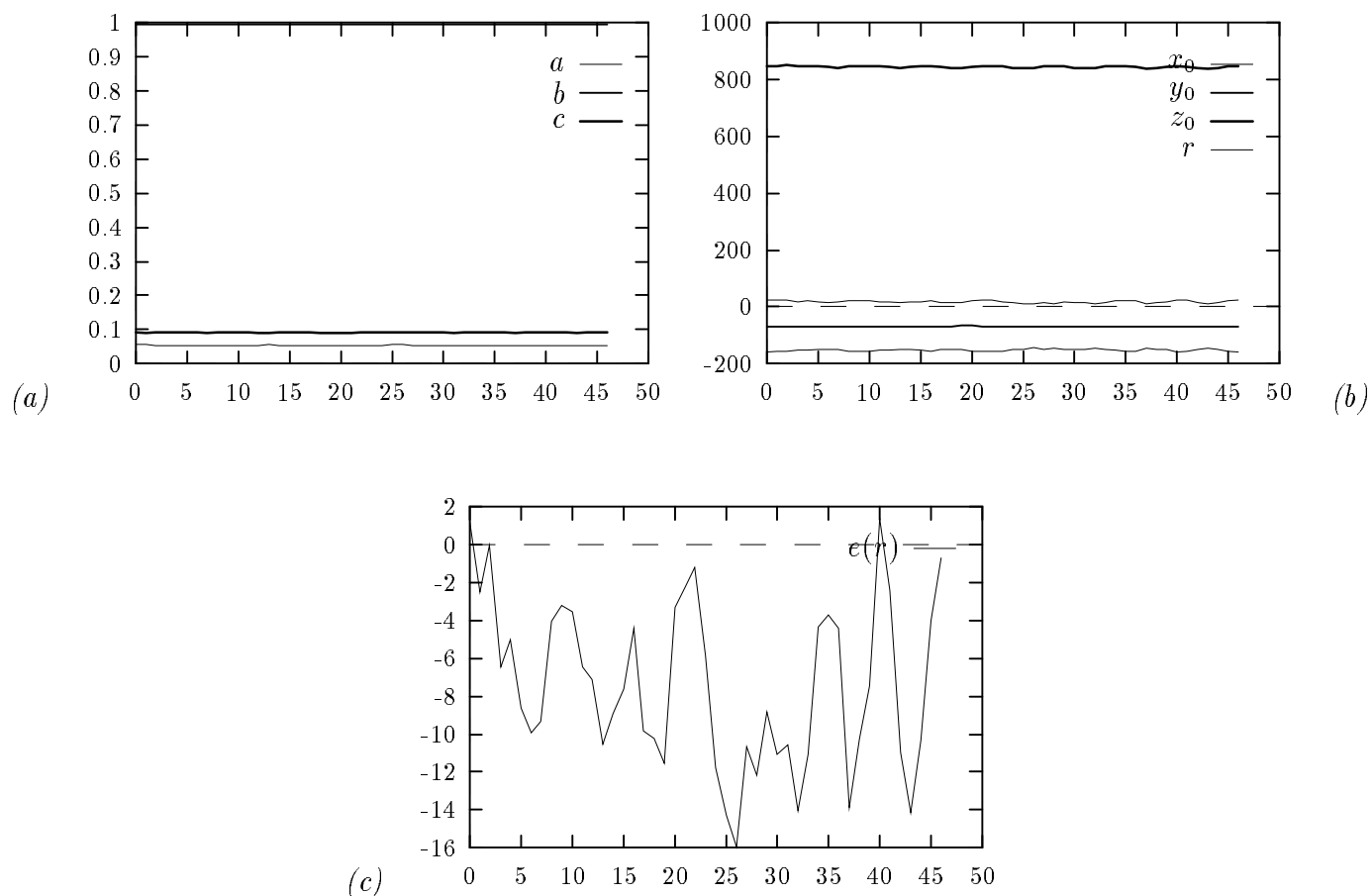


Figure 5.28 : *Reconstruction 3D du second cylindre de la scène par vision active en utilisant la méthode basée sur un seul limbe (déplacement important) : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r (en mm). (c) erreur entre la valeur estimée du rayon et sa valeur réelle (25 mm).*

5.4.2 Méthode de reconstruction 3D basée sur deux limbes

Le déclenchement du processus de reconstruction 3D, intégrant les informations 2D associées aux deux limbes du cylindre, nécessite les trois phases suivantes :

- une phase d'initialisation qui consiste à acquérir une première estimée grossière de la structure 3D du cylindre sélectionné en utilisant la méthode de reconstruction 3D basée sur un seul limbe, celui-ci étant suivi d'image à image.
- une phase de prédiction qui consiste à tirer parti de l'estimation obtenue à l'étape précédente afin de prédire la position du second limbe dans l'image.
- une phase de focalisation qui consiste à réinitialiser le processus de suivi qui, cette fois, va prendre en charge simultanément les deux segments correspondant à la projection des deux limbes du cylindre dans l'image.

Connaissant, à chaque nouvelle acquisition d'images, les paramètres 2D de la projection dans l'image des deux limbes du cylindre, on peut alors utiliser la méthode de reconstruction 3D adéquate.

5.4.2.1 Reconstruction 3D d'un cylindre par vision dynamique

Les résultats issus d'un mouvement quelconque (toujours le même que dans les cas précédents) en partant de la position initiale de la figure 5.18 sont présentés sur la figure 5.29 qui est configurée comme la figure 5.19. La figure 5.30 illustre quant à elle l'évolution des paramètres (ρ_1, θ_1) , (ρ_2, θ_2) correspondant respectivement à chacun des limbes du cylindre. Comme on pouvait facilement le prévoir, ces résultats sont moins biaisés que ceux obtenus en utilisant la méthode basée sur un seul limbe (voir figure 5.19). Ces résultats sont cependant insatisfaisants et nous allons voir qu'on peut grandement les améliorer en utilisant notre approche en boucle fermée.

5.4.2.2 Reconstruction 3D d'un cylindre par vision active

La position optimale d'un cylindre est choisie telle qu'il apparaisse sous la forme de deux droites verticales symétriques dans l'image. La figure 5.31 illustre la position optimale du cylindre dans l'image. Le mouvement optimal consiste alors, une fois la convergence établie, à garder en permanence ce motif constant dans l'image. La figure 5.32 montre la convergence parfaite de la loi de commande vers la configuration optimale désirée du cylindre dans l'image.

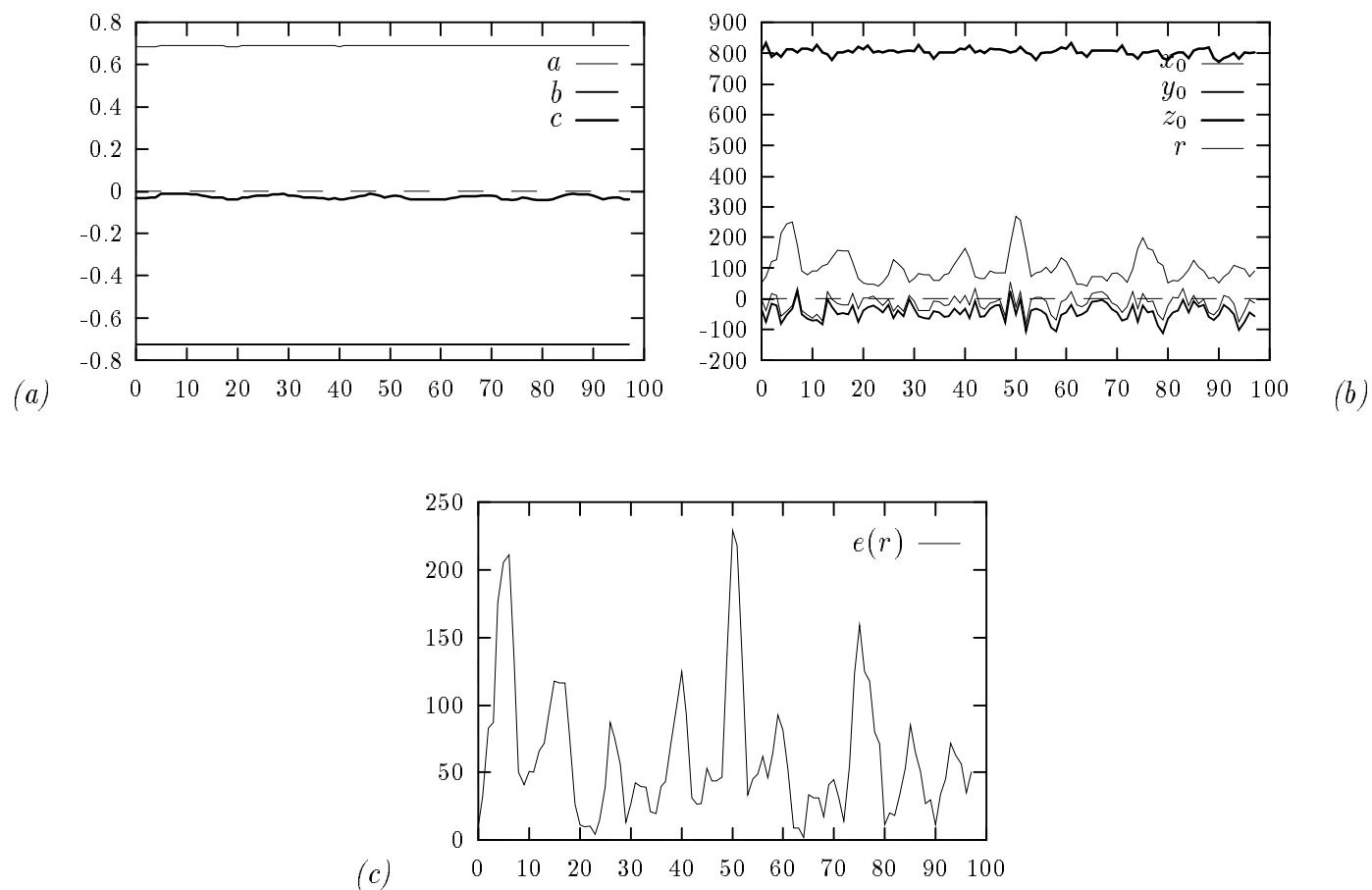


Figure 5.29 : Reconstruction 3D du premier cylindre sélectionné de la scène par vision dynamique en utilisant la méthode basée sur deux limbes : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r (en mm). (c) erreur entre la valeur estimée du rayon et sa valeur réelle (40 mm).

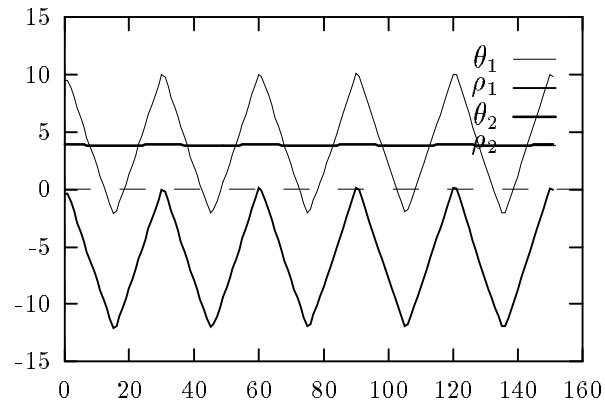


Figure 5.30 : Paramètres 2D des limbes du premier cylindre sélectionné de la scène sans stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées polaires des deux droites correspondant à la projection du cylindre dans l'image.



Figure 5.31 : Position optimale entre la caméra et le premier cylindre sélectionné de la scène

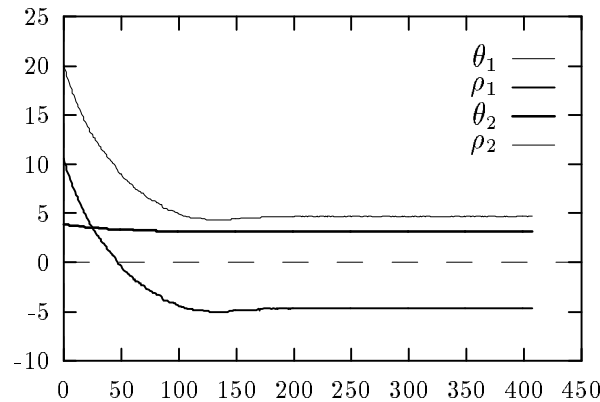


Figure 5.32 : Paramètres 2D du premier cylindre de la scène avec une stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées polaires des deux droites correspondant à la projection du cylindre dans l'image.

Les résultats de reconstruction obtenus après la phase de convergence sont présentés sur la figure 5.33. Ces résultats sont, encore une fois, nettement plus précis que dans le cas précédent où un seul limbe du cylindre était utilisé. On peut noter que l'erreur commise sur l'estimation du rayon est inférieure à $1/2$ mm, ce qui représente une amélioration très notable comparée aux erreurs relativement importantes commises sur l'estimation du rayon en utilisant la méthode basée sur un seul limbe (voir figure 5.23). Comme dans les deux cas précédents du point et de la droite, on peut remarquer que quelques itérations suffisent (voir l'erreur observée sur le rayon) pour obtenir une bonne estimation des paramètres du cylindre. On peut noter aussi que la qualité de ces résultats est très supérieure à celle obtenue par vision dynamique (voir 5.29). Ceci confirme les conclusions que nous avons tirées en nous basant sur les résultats de simulation obtenus au paragraphe 2.3.6, à savoir que la vision dynamique ne se suffit pas à elle-même pour atteindre des performances acceptables en reconstruction 3D. Il est clair que, les algorithmes de reconstruction 3D étant très liés au mouvement de la caméra, ce mouvement doit être judicieusement choisi de manière à se placer dans des conditions favorables au processus de reconstruction 3D.

La figure 5.34 montre les résultats de reconstruction 3D obtenus sur ce même cylindre en utilisant cette fois un déplacement important de la caméra entre les deux images prises en compte pour la reconstruction. Contrairement au cas d'un seul limbe, on note seulement une légère amélioration par rapport au cas précédent. Ceci est justifié par le fait que dans le cas d'un seul limbe, un fort déplacement permet de compenser en quelque sorte l'absence d'informations sur le second limbe alors

que dans le cas présent, il contribue uniquement à l'amélioration de la robustesse de l'estimation. Le tableau 5.4 présente les moyennes et les écarts types qui ont été évalués sur les paramètres 3D estimés du cylindre considéré. Sur ce tableau, on peut noter une fois de plus la stabilité des résultats obtenus et l'excellente précision sur l'estimation du rayon du cylindre.

	Moyenne	Ecart type
a	-0.67	0.0009
b	0.73	0.0008
c	0.036	0.002
x_0	-86.5 mm	1.3
y_0	-121.5 mm	1.1
z_0	850.8 mm	1.1
r	40.03 mm	0.14

Tableau 5.4 : Statistiques sur les paramètres 3D estimés du premier cylindre de rayon 40 mm (où a, b et c définissent l'orientation du cylindre, x_0, y_0, z_0 sa position et r son rayon)

Enfin, en utilisant la même stratégie optimale de reconstruction 3D pour le second cylindre de la scène, on obtient des résultats similaires à ceux qui viennent d'être présentés. La figure 5.35 illustre la position finale obtenue après la phase de convergence entre la caméra et le cylindre en partant de la position initiale de la figure 5.25. Sur la figure 5.36 est montrée l'évolution des paramètres 2D associés au cylindre suivi. Les résultats de reconstruction 3D obtenus sont présentés sur la figure 5.37, les moyennes et les écarts types correspondant à ces estimations étant récapitulés sur le tableau 5.5. On peut constater une fois de plus la robustesse de notre approche en boucle fermée.

	Moyenne	Ecart type
a	-0.67	0.0006
b	0.74	0.0005
c	0.038	0.002
x_0	-90.5 mm	1.6
y_0	-127.9 mm	1.6
z_0	870.6 mm	1.4
r	24.68 mm	0.09

Tableau 5.5 : Statistiques sur les paramètres 3D estimés du second cylindre de rayon 25 mm (où a, b et c définissent l'orientation du cylindre, x_0, y_0, z_0 sa position et r son rayon)

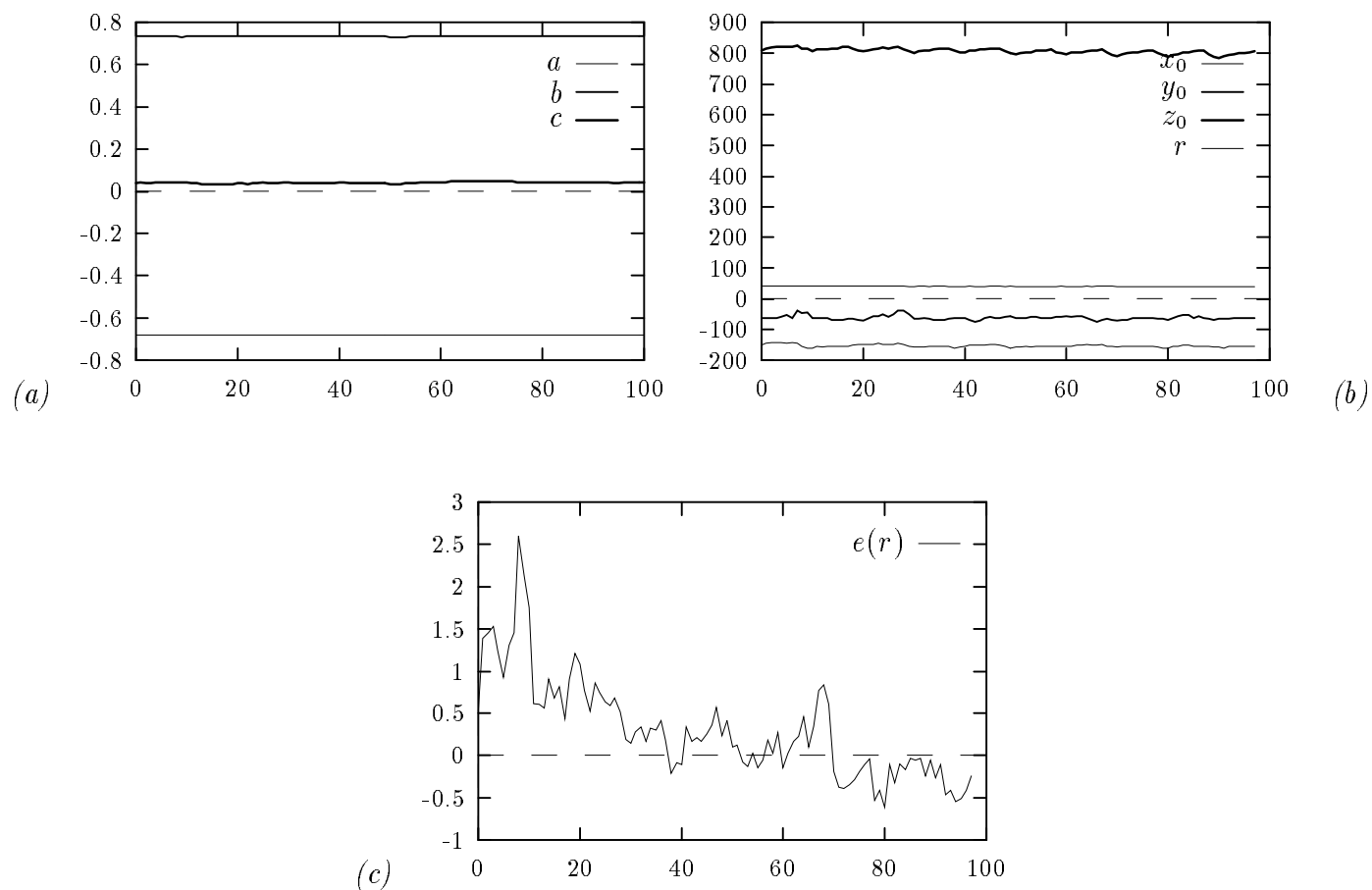


Figure 5.33 : Reconstruction 3D du premier cylindre de la scène par vision active en utilisant la méthode basée sur deux limbes (faible déplacement) : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r (en mm). (c) erreur entre la valeur estimée du rayon et sa valeur réelle (40 mm).

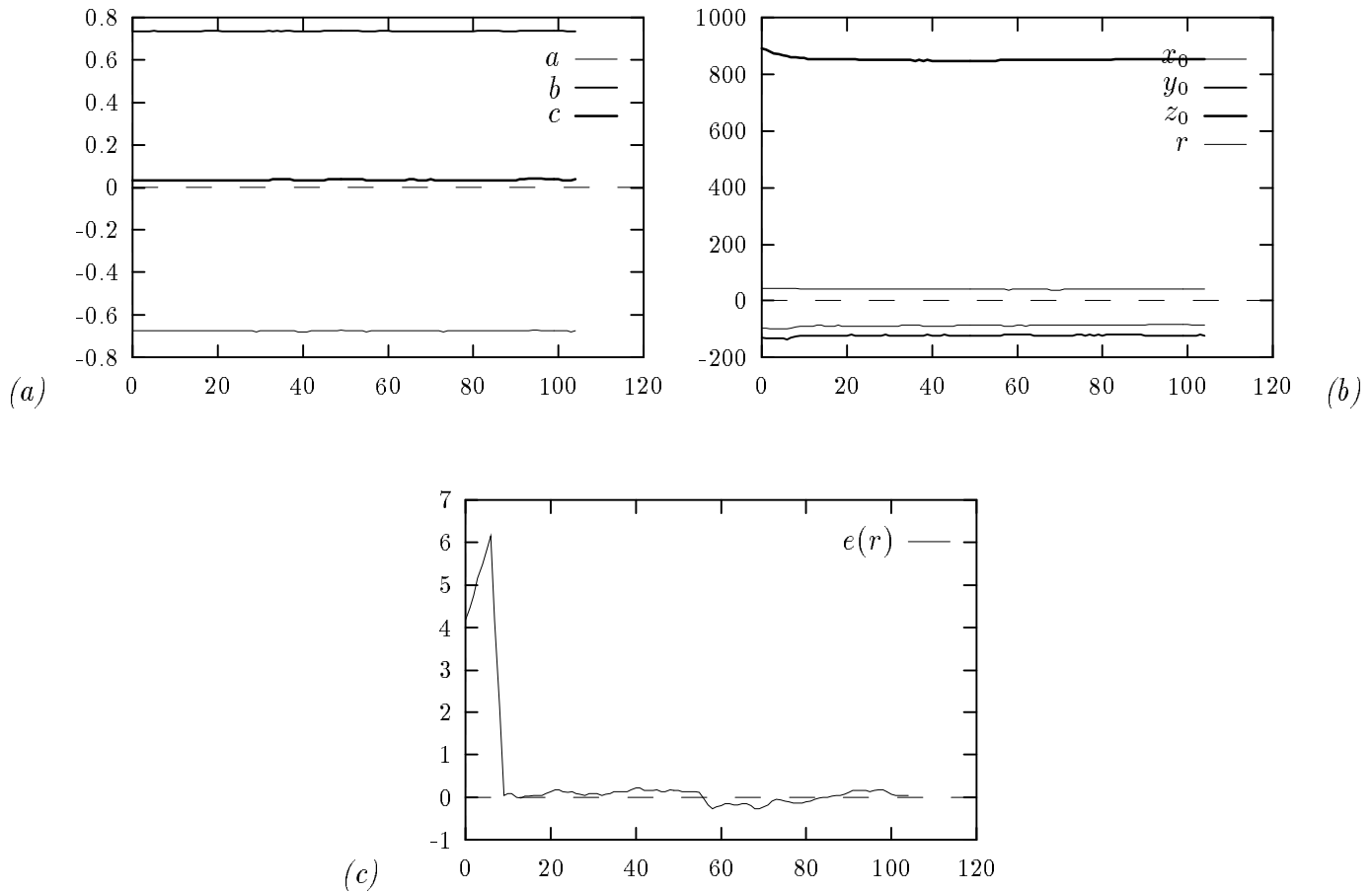


Figure 5.34 : *Reconstruction 3D du premier cylindre de la scène par vision active en utilisant la méthode basée sur deux limbes (déplacement important) : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r (en mm). (c) erreur entre la valeur estimée du rayon et sa valeur réelle (40 mm).*



Figure 5.35 : *Position optimale entre la caméra et le second cylindre sélectionné de la scène.*

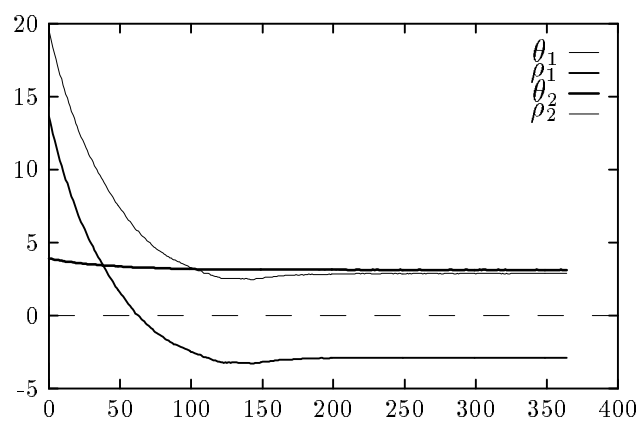


Figure 5.36 : *Paramètres 2D du second cylindre de la scène avec une stratégie optimale pour le mouvement de la caméra : valeurs mesurées à chaque itération des coordonnées polaires des deux droites correspondant à la projection du cylindre dans l'image.*

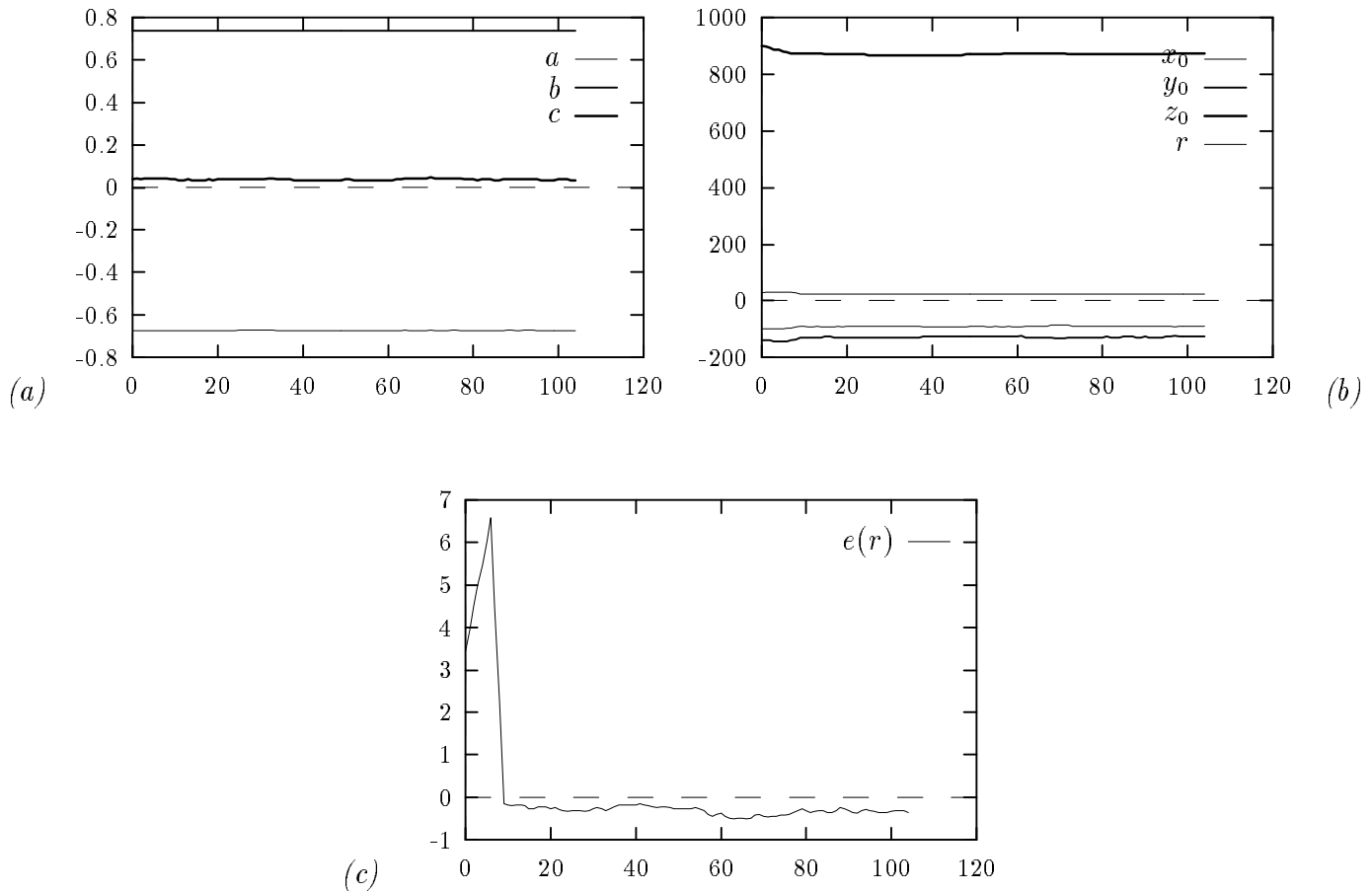


Figure 5.37 : Reconstruction 3D optimale du second cylindre de la scène en utilisant la méthode basée sur deux limbes (déplacement important) : (a) valeurs estimées à chaque itération des paramètres a, b et c définissant l'orientation du cylindre. (b) valeurs estimées à chaque itération des paramètres x_0, y_0, z_0 définissant la position du cylindre et de son rayon r (en mm). (c) erreur entre la valeur estimée du rayon et sa valeur réelle (25 mm).

Conclusion

Le travail présenté dans cette thèse a porté sur la conception d'un module de perception tridimensionnelle de l'environnement. Ce problème a été abordé dans le cadre d'une observation monoculaire d'un environnement statique à l'aide d'une caméra mobile commandable.

Dans un premier temps, nous nous sommes attachés à établir des méthodes de reconstruction 3D par vision dynamique. Nous avons en effet proposé une méthode générale de reconstruction 3D de primitives géométriques paramétrables s'appuyant sur un formalisme élégant et unifié : le torseur d'interaction associé aux primitives de l'environnement caractérisant complètement les interactions entre la caméra et les primitives considérées. Ce formalisme, parfaitement adapté aux problèmes de perception 3D, s'applique aisément aussi bien aux objets polyédriques qu'aux objets non polyédriques tels que les cylindres, les sphères, etc.

Nous avons ensuite cerné les limitations de ce système de reconstruction 3D et justifié alors le souci de le doter d'une perception plus "intelligente" qui permettrait de mieux percevoir l'environnement.

Ces limitations sont principalement :

- le biais induit par le phénomène de discrétisation du fait de l'aspect continu de l'approche.
- sa sensibilité au bruit, problème majeur rencontré par toute méthode de reconstruction 3D.

Ce travail a alors souligné l'importance de maîtriser le mouvement de la caméra en perception 3D. A notre sens, la vision dynamique, bien qu'elle se soit montrée nettement plus performante que la vision monoculaire passive, ne peut se suffire à elle-même pour affronter le problème de la reconstruction 3D dans toute sa complexité. Une perception intelligente de l'environnement, en l'occurrence la vision active, est plus appropriée et plus en mesure de faire face aux multiples difficultés rencontrées dans un monde réel (bruit, occlusion,..).

Dans un second volet de notre travail, nous avons donc abordé le problème de la reconstruction 3D des primitives de l'environnement en nous plaçant dans le contexte bien plus favorable d'un observateur actif dont les mouvements sont contrôlés de façon à :

- supprimer l'erreur de discrétisation;
- minimiser les erreurs de reconstruction dues aux imprécisions de mesure.

Ces mouvements sont calculés de manière automatique à l'aide d'un schéma robuste de commande en boucle fermée sur les informations visuelles. Nous avons ainsi conçu un processus de reconstruction adaptatif permettant constamment d'améliorer l'estimation de la structure 3D des primitives au cours du temps.

Une troisième partie de ce travail, non moins importante que les deux précédentes, a été consacrée aux aspects de traitement d'images. Un module d'extraction de caractéristiques visuelles et de leur suivi dans une séquence d'images est naturellement un maillon indispensable à tout système de perception 3D. A cet effet, nous avons proposé une méthode robuste et originale de suivi de segments dans une séquence d'images. C'est une approche locale intégrant un algorithme de calcul d'éléments de contour en mouvement dont une version améliorée a été conçue. Cette méthode présente de nombreux avantages dont :

- sa robustesse aux aléas de segmentation et d'occlusion;
- son implémentation aisée sur des architectures parallèles;
- sa généralité et sa versatilité.

Afin de satisfaire les contraintes de temps calcul imposées par notre approche de reconstruction 3D en boucle fermée, cet algorithme été porté sur la carte Sympati2, une carte multiprocesseurs adaptée au traitement d'images. Cela a permis

d'atteindre des performances de calcul quasi temps réel. Des performances très intéressantes ont été aussi obtenues sur la carte EDIXIA-1000 dont nous disposons à l'IRISA.

Outre ses performances temps réel, cet algorithme s'est montré très efficace lors de sa validation sur plusieurs séquences d'images.

Enfin, dans la dernière partie de cette thèse, nous nous sommes attachés à évaluer l'intérêt de notre système actif de reconstruction 3D en l'implantant sur un site expérimental composé d'une caméra embarquée sur un robot cartésien à six degrés de liberté. Plusieurs expérimentations portant notamment sur la reconstruction de primitives point, droite et cylindre ont prouvé la validité de notre système et ont montré que l'on pouvait obtenir des résultats de reconstruction fiables et précis, bien meilleurs que ceux obtenus en utilisant la vision dynamique. On rejoint ainsi le point de vue des travaux antérieurs [BAN86], [ALO87], [BAJ88], [SAN90], [TIS92], [BAL91] soulignant l'apport incontestable de la vision active dans différents aspects de la vision.

Ce travail a révélé de nombreuses perspectives de recherche qui n'ont pu être couvertes par cette thèse. Il serait souhaitable en effet d'approfondir les questions suivantes :

- l'identification des primitives à reconstruire. En effet, dans le travail que nous avons présenté, nous avons supposé connu le type des primitives à reconstruire;
- la prise en compte d'objets plus complexes que les droites, les sphères ou les cylindres;
- l'élaboration de stratégies globales pour la reconstruction optimale de scènes complexes constituées de plusieurs primitives.

Sur ce dernier aspect, le travail a d'ores et déjà débuté au sein de notre équipe [MAR93]. Quelques investigations ont permis d'élaborer une méthode de reconstruction 3D d'une scène complexe, intégrant le module de reconstruction 3D optimale que nous avons présenté dans cette thèse. Cette méthode est mise en œuvre par un réseau d'automates qui gère automatiquement l'enchaînement des différentes phases liées à la reconstruction globale de la scène perçue, à savoir :

- la sélection des primitives à reconstruire. Un ensemble de segments extraits de l'image initiale constitue la base de données 2D initiale qui va être utilisée

pour déterminer l'ordre de reconstruction des primitives. Cette base est mise à jour au fur et à mesure du processus de reconstruction 3D;

- la reconstruction précise une à une de ces primitives à l'aide de l'approche vision active que nous avons proposée dans ce travail;
- simultanément, la reconstruction 3D sommaire (par vision dynamique) des autres primitives présentes dans le champ visuel de la caméra. L'objectif de cette phase est d'acquérir des informations complémentaires sur la scène de manière à optimiser les phases d'exploration et à faciliter le calcul des trajectoires de la caméra en environnement incertain.

Après chaque reconstruction, le robot se replace à sa position de départ de manière à observer la scène correspondant à la base de données 2D en mémoire afin de reconstruire la primitive suivante.

Une phase d'exploration sera déclenchée lorsque tous les segments de la base de données 2D seront associés à une primitive de la scène. Le but de l'exploration est de découvrir des zones de la scène dont les primitives n'ont pas encore été reconstruites. Des prises de vue appropriées, calculées à l'aide de la triangulation de Delauney, permettent d'atteindre la complétude de la reconstruction 3D de la scène. A la fin de chaque phase d'exploration, une base de données 2D est construite et le processus de reconstruction optimale de la scène est déclenché. Ce processus prend fin quand la scène a été entièrement reconstruite.

Annexe A

Les transformations géométriques

Etant donnés deux repères orthonormés $(O_i, \vec{x}_i, \vec{y}_i, \vec{z}_i)$ et $(O_j, \vec{x}_j, \vec{y}_j, \vec{z}_j)$, la situation de R_j par rapport à R_i est représentée par la matrice homogène M_i^j :

$$M_i^j = \begin{pmatrix} R_i^j & T_i^j \\ 0 & 1 \end{pmatrix} \quad (\text{A.1})$$

T_i^j étant le vecteur de translation correspondant aux coordonnées de O_j dans R_i et R_i^j la matrice de rotation entre $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ et $(\vec{x}_j, \vec{y}_j, \vec{z}_j)$.

Parmi les propriétés de ces matrices homogènes, on peut rappeler :

- $M_i^j M_j^k = M_i^k$
- $M_j^i = M_i^j{}^{-1} = \begin{pmatrix} R_i^j{}^T & -R_i^j{}^T T_i^j \\ 0 & 1 \end{pmatrix}$.

Nous avons représenté les rotations sous la forme (\underline{u}, θ) où $\underline{u} = (u_x, u_y, u_z)$ est l'axe normé de la rotation et θ l'angle de rotation autour de cet axe. Le passage de cette représentation à la représentation matricielle est immédiat. En effet, la matrice de rotation R correspondant à la représentation (\underline{u}, θ) se déduit directement à partir de la relation suivante :

$$R = \mathbb{I}_3 \cos \theta + (1 - \cos \theta) \underline{u}\underline{u}^T + A(\underline{u}) \sin \theta \quad (\text{A.2})$$

avec \mathbb{I}_3 la matrice identité et $A(\underline{u})$ la matrice antisymétrique :

$$A(\underline{u}) = \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix}$$

Réciproquement, on peut facilement déterminer la représentation (\underline{u}, θ) correspondant à une matrice de rotation R en utilisant les deux relations suivantes :

$$\begin{cases} \cos \theta &= \frac{1}{2}(R_{11}R_{22}R_{33} - 1) \\ A(\underline{u}) \sin \theta &= \frac{1}{2}(R - R^T) \end{cases} \quad (\text{A.3})$$

R_{11}, R_{22} et R_{33} étant les coefficients diagonaux de la matrice R .

En se fixant θ positif, \underline{u} est déterminé de manière unique si $\sin \theta \neq 0$.

Soit $T_c = (V_c, \Omega_c)$ le torseur cinématique calculé par la loi de commande et appliqué à la caméra, V_c étant la vitesse de translation de la caméra et Ω_c étant sa vitesse de rotation. T_c est choisie comme la vitesse moyenne de la caméra, calculée à partir de la mesure des positions de la caméra à chaque acquisition d'image ($\dot{T} = 0$).

Soit $m = \underline{x}^T$ un point immobile de la scène. La vitesse du point m exprimée dans le repère caméra s'écrit :

$$\dot{\underline{x}} = -V_c - \Omega_c \wedge \underline{x} \quad (\text{A.4})$$

En utilisant une approximation au premier ordre on obtient la matrice homogène de déplacement suivante :

$$M = \begin{pmatrix} R_c & V_c \\ 0 & 1 \end{pmatrix} \quad (\text{A.5})$$

avec $R_c = \cos \theta \mathbb{I}_3 + (1 - \cos \theta) \underline{u} \cdot \underline{u}^T + \sin \theta \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix}$

où $\theta = \|\Omega_c\|$ et $\underline{u}^T = (u_1 \ u_2 \ u_3) = \Omega_c / \|\Omega_c\|$.

Annexe B

Le calibrage

B.1 Introduction

La reconstruction 3D des primitives de l'environnement nécessite la détermination préalable des transformations géométriques liant le plan image à l'espace tridimensionnel. En effet, l'interprétation des mesures de la structure 3D des objets de la scène perçue par la caméra se fait par rapport à un repère de référence différent du repère de projection. Il faut donc positionner ces deux repères l'un par rapport à l'autre, autrement dit déterminer le déplacement rigide liant le repère de la caméra au repère 3D de référence. De plus, la caméra étant un capteur, il est nécessaire de l'étalonner pour corriger toutes les déformations qu'elle engendre. Ces deux classes de paramètres sont définies de la façon suivante :

- les paramètres intrinsèques de la caméra qui dépendent de ses caractéristiques électroniques et physiques.
- les paramètres extrinsèques de la caméra qui dépendent de sa position et de son orientation dans un repère du monde.

L'identification de ces paramètres s'effectue par une procédure de calibrage. C'est une opération qui consiste à déterminer la transformation liant les coordonnées 3D d'un point objet et les coordonnées 2D du point image correspondant. Cette transformation est souvent exprimée sous forme d'une matrice (3×4) dite matrice de transformation perspective [TSA86], [TOS87], [CHA89].

Une bonne revue critique sur les principales méthodes de calibration existantes est présentée dans [TSA86].

Pour se ramener à des équations linéaires, le passage aux coordonnées homogènes est nécessaire. Les deux équations précédentes peuvent s'écrire, en posant $F_i = fK_i$ et $F_j = fK_j$, sous la forme suivante :

$$\begin{pmatrix} Si \\ Sj \\ S \end{pmatrix} = \begin{pmatrix} F_i & 0 & i_c & 0 \\ 0 & F_j & j_c & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (\text{B.7})$$

Les observations étant les coordonnées (x_o, y_o, z_o) de points 3D m exprimées dans un repère objet choisi R_o , on doit s'y ramener grâce à la matrice de passage $M_c^\circ = (R_c^\circ \ T_c^\circ)$ liant le repère objet au repère caméra, R_c° étant la matrice de rotation et T_c° le vecteur de translation. Les paramètres représentant cette matrice sont appelés les paramètres extrinsèques de la caméra.

Le système précédent peut alors s'exprimer par la relation suivante :

$$\underline{u} = M\underline{x} \quad (\text{B.8})$$

où $\underline{u} = (Si, Sj, S)^T$, $\underline{x} = (x, y, z)^T$ et M est une matrice (3×4) dite matrice de transformation perspective. Elle est définie à un facteur d'échelle près (ω).

$$M = \omega \begin{pmatrix} F_i r_1 + i_c r_3 & F_i t_x + i_c t_z \\ F_j r_2 + j_c r_3 & F_j t_y + j_c t_z \\ r_3 & t_z \end{pmatrix} \quad (\text{B.9})$$

où r_i représente le i^{eme} vecteur ligne de la matrice de rotation R_c° .

B.3 Distorsion de la lentille

Il existe principalement deux types de distorsion de la lentille :

- la distorsion radiale;
- la distorsion tangentielle.

En général, dans les processus industriels, seule la distorsion radiale est prise en considération. Cette dernière est modélisée par une série infinie dont on ne garde souvent uniquement deux termes [TSA86]. La position du point m réellement observée dans l'image est le point $m_d = (X_d, Y_d)^T$ tel que :

$$\begin{cases} X_d = X + D_x \\ Y_d = Y + D_y \end{cases} \quad (\text{B.10})$$

avec :

$$\begin{cases} D_x &= X(k_1 r^2 + k_2 r^4) \\ D_y &= Y(k_1 r^2 + k_2 r^4) \\ r &= X^2 + Y^2 \end{cases} \quad (\text{B.11})$$

B.4 Estimation des paramètres de la caméra

La plupart des méthodes existantes supposent les coefficients de distorsion nuls afin de simplifier le problème et de le formuler en terme d'un système linéaire [TOS87]. En ce qui nous concerne, nous avons utilisé l'approche présentée dans [CHA89] qui consiste en deux étapes. Une estimation grossière des paramètres de la caméra est d'abord déterminée à l'aide d'une technique de résolution linéaire. Ensuite, ces valeurs sont utilisées pour initialiser un algorithme itératif d'optimisation non linéaire des paramètres de la caméra, incluant les coefficients de distorsion.

Cette approche présente l'avantage de supprimer le biais qui apparaît sur les paramètres issus de la méthode linéaire et aussi de modéliser les distorsions géométriques introduites par l'objectif de la caméra.

Une fois que les paramètres intrinsèques de la caméra ont été déterminés, on peut alors utiliser les équations de projection perspective qui sont à la base même de notre approche de reconstruction 3D. Dans le cas du point, les mesures étant les coordonnées (i, j) , exprimées en pixel, de la projection dans l'image du point 3D, le changement de variable suivant est effectué, (X, Y) étant les coordonnées du point 2D exprimées en *mm* :

$$\begin{cases} X &= F_i(i - i_c) \\ Y &= F_j(j - j_c) \end{cases} \quad (\text{B.12})$$

De même, le changement de variable qui s'impose pour la reconstruction 3D d'une droite est :

$$\begin{cases} \rho_t &= \frac{\rho - i_c \cos \theta - j_c \sin \theta}{\sqrt{F_i^2 (\cos \theta)^2 + F_j^2 (\sin \theta)^2}} \\ \theta_t &= \arctan(F_j \sin \theta, F_i \cos \theta) \end{cases} \quad (\text{B.13})$$

avec :

- ρ (exprimée en *pixels*) et θ les coordonnées polaires de la projection de la droite dans l'image,
- ρ_t (exprimée en *mm*) et θ_t les valeurs transformées correspondantes en utilisant les paramètres obtenus par calibrage.

Annexe C

Le système multiprocesseurs Sympati2

C.1 Introduction

Sympati2 est un processeur ligne. La structure de base est un anneau constitué de m processeurs élémentaires (PEs) et m bancs mémoire, avec m pouvant varier en puissance de 2, de 16 à 256 PEs. L'ensemble PEs et bancs mémoire fonctionnent en mode SIMD (Single Instruction Multiple Data) sous le contrôle d'une unité de contrôle programmable. Le mode SIMD signifie que tous les processeurs élémentaires effectuent, de façon synchrone, les mêmes opérations sur des données différentes. Ce système est commercialisé par Centralp Automatismes sous le nom "openvision". C'est un système qui peut être utilisé soit comme module d'acquisition et d'accélération de traitement d'images connecté à une station de travail par liaison ETHERNET TCP/IP, soit comme module de vision intégré dans une application temps réel.

C.2 Architecture de Sympati2

Sympati2 est composé de deux blocs : une unité de commande (UC) et un bloc de calcul (16 à 256 processeurs élémentaires). Le rôle de l'UC est d'alimenter en instructions les PEs et d'assurer la communication entre le calculateur hôte et le processeur ligne. Le bloc calcul est formé de l'ensemble des PEs et de leurs bancs mémoire associés. Le PE se décompose en trois parties : une partie traitement, une partie adressage et la partie interconnexion [ESS88], [BAS88].

- la partie traitement est constituée d'une unité arithmétique et logique (ALU) de 16 bits, d'un multiplicateur de 8 bits, d'un module de décalage et d'un ensemble de registres de travail.
- la partie adressage permet deux types d'adressage : un adressage hélicoïdal de la mémoire, où une adresse commune est envoyée à tous les PEs par l'UC, les PEs calculant leur adresse par référence à cette donnée commune et un adressage tabulaire où chaque PE accède à un des bancs mémoire à partir d'une adresse engendrée par la partie traitement.
- la partie interconnexion permet l'accès des PEs à un grand voisinage de l'image.

Notons que le segment de PEs peut être choisi horizontal ou vertical. Dans les deux cas, deux types de balayage peuvent être utilisés, le balayage linéaire et le balayage par bandes [ADA89].

C.2.1 Arrangement hélicoïdal des données

La matrice de données est distribuée de manière hélicoïdale dans les bancs mémoire des PEs pour que les points d'un segment ligne ou colonne se trouvent dans des bancs mémoire distincts, à la même adresse. Ceci, afin d'éviter les conflits d'accès de m points consécutifs à une ligne ou une colonne de la matrice (m étant le nombre de PEs). La figure (C.1) illustre un exemple d'arrangement hélicoïdal des données d'une image (8*8) dans 4 PEs (le segment est alors formé de 4 pixels).

Chaque grisé représente le banc mémoire d'un PE.

Parfois, ce type d'arrangement n'est pas parfaitement adapté notamment à certains algorithmes de traitement d'images de moyen ou de haut niveau. Sympati2 offre donc un autre mode d'arrangement des données : le mode tabulaire dans lequel la matrice de données est divisée en bandes (voir figure C.2).

Avec une machine composée de 32 PEs et de bancs mémoire de 32 Ko, la mémoire peut contenir une image 1024*1024 pixels. On obtient 32 bandes de 32 pixels. Chaque bande est divisée en 32 sous-bandes, soit une sous-bande par PE (1*1024).

C.2.2 Interconnexion entre les PEs

La partie interconnexion est constituée d'un ensemble de chemins de données permettant, d'une part, aux différents PEs de communiquer entre eux, et, d'autre part,

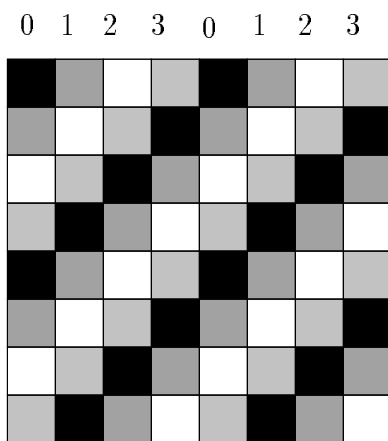


Figure C.1 : Arrangement hélicoïdal des données

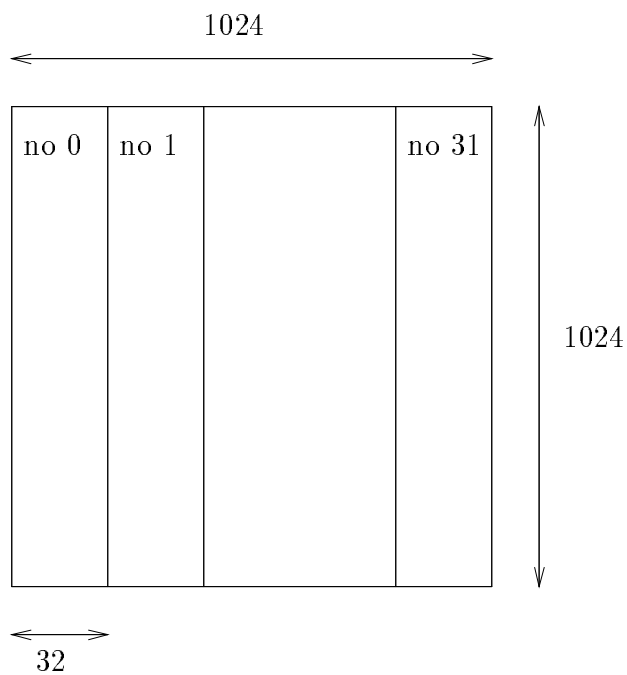


Figure C.2 : Découpage en bandes de l'image avec 32 PEs

de transférer les données entre les différents modules (bancs mémoires, PEs et UC). On distingue donc deux types de transfert :

- transferts entre les PEs : chaque PE peut effectuer des échanges (chargement de registres) avec les PEs de gauche et de droite à une distance maximale de trois.
- transferts entre les PEs et les bancs mémoires : chaque PE peut lire dans les bancs mémoire des PEs de gauche et de droite à une distance maximale de 2.

C.3 Environnement de programmation

L'environnement de programmation est constitué d'un langage parallèle, le langage 4LP (Low Level Language for Line Processor), de son compilateur et d'un debugger. Le langage 4LP est un langage conçu pour une structure SIMD, dont le parallélisme est totalement transparent pour le programmeur. La philosophie du langage consiste à focaliser le traitement sur un pixel de l'image. Ensuite, le travail est répété pour l'ensemble des pixels de l'image. Pour plus de précisions sur ce langage le lecteur est prié de se référer au manuel d'utilisation de 4LP [ADA89].

C.3.1 Application au traitement d'images

Les traitements privilégiés sur la machine Sympati2 sont les traitements d'images de bas et moyen niveaux, donc l'ensemble des opérations qui s'effectuent au niveau pixel. Les traitements de moyen niveau correspondent aux algorithmes qui extraient des informations de l'image (contours, textures,...). Par contre la machine est beaucoup moins performante pour des algorithmes de haut niveau car ils sont souvent par nature même séquentiels.

Bibliographie

- [ADA89] ADAM P. Manuel d'utilisation du langage assembleur de la machine sympati2. *Rapport Interne CEA*, No. 88.34, Août 1989.
- [ADI85] ADIV G. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on PAMI*, Vol. 7, No. 4:pp. 384–401, Juillet 1985.
- [ADI89] ADIV G. Inherent ambiguities in recovering 3d motion and structure from a noisy flow field. *IEEE Trans. on PAMI*, Vol. 11, No. 5:pp. 477–489, Mai 1989.
- [AGG87] AGGARWAL J.K., WANG Y.F. Analysis of a sequence of images using point and line correspondences. *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1275–1280, Mars 1987.
- [AGG88] AGGARWAL J.K., NANDHAKUMAR N. On the computation of motion from sequences of images. a review. *Proc. of the IEEE*, Vol. 76, No. 8:pp. 917–935, Août 1988.
- [ALO87] ALOIMONOS J., WEISS I., BANDOPADHAY .A. Active vision. 1st *Int. Conf. on Computer Vision, London, England*, pp. 35–54, Juin 1987.
- [ALO88] ALOIMONOS J. Visual shape computation. *Proc. of IEEE*, Vol. 76, No. 8:pp. 899–916, 1988.
- [ALO90] ALOIMONOS J. Purposive and qualitative active vision. *Proc. on 10th ICPR, New Jersey*, Vol. 1:pp. 346–360, 1990.
- [ARB91] ARBOGAST E. Modélisation automatique d'objets non polyédriques par observation monoculaire. *Thèse de l'Institut National Polytechnique de Grenoble*, Juillet 1991.

- [AYA89a] AYACHE N. *Vision stéréoscopique et perception multisensorielle. Applications à la robotique mobile*. Collection Science Informatique, Inter-Editions, Paris, 1989.
- [AYA89b] AYACHE N., FAUGERAS O.D. Maintaining representation of the environment of a mobile robot. *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 6:pp. 804–819, Décembre 1989.
- [BAJ88] BAJCSY R. Active perception. *Proc. of the IEEE*, Vol. 76, No. 8, pp. 996–1005, Août 1988.
- [BAL91] BALLARD D.H. Animate vision. *Artificial Intelligence*, Vol. 48:pp. 57–86, 1991.
- [BAN86] BANDOPADHAY .A, CHANDRA B., BALLARD D.H. Egomotion using active vision. *Int. Conf. on CVPR 86*, pp. 498–503, Juin 1986.
- [BAS88] BASILLE J.L., ESSAFI H., JUVIN D., LATIL J.Y. Sympati2 : a 1.5D processor array for image application. *Proc. 4th Conf. EUSIPCO, Grenoble*, 1988.
- [BOU87] BOUTHEMY P. Estimation et structuration d'indices spatio-temporels pour l'analyse du mouvement dans une séquence d'images. *Traitement du Signal*, Vol. 4, No. 3:pp. 239–257, 1987.
- [BOU89a] BOUKARRI B. Reconstruction 3D récursive de scènes structurées au moyen d'une caméra mobile. Application à la robotique. *Thèse de l'Université d'Orsay*, Octobre 1989.
- [BOU89b] BOUTHEMY P. A maximum likelihood framework for determining moving edges. *IEEE Trans. on PAMI*, Vol. 11, No. 5:pp. 499–511, Mai 1989.
- [BOU92a] BOUKIR S., BOUTHEMY P., CHAUMETTE F., JUVIN D. Mise en correspondance de segments dans une séquence d'images par une approche locale. *Rapport de recherche IRISA*, No. 685, 28 pages, Octobre 1992.
- [BOU92b] BOUKIR S., CHAUMETTE F. Reconstruction 3D de primitives géométriques paramétrables par vision active. *Rapport de recherche INRIA*, No. 1722, 38 pages, Juillet 1992.
- [BOU93a] BOUKIR S., BOUTHEMY P., CHAUMETTE F., JUVIN D. Real-time contour matching over time in an active vision context. *8th SCIA, Tromso, Norvège*, Mai 1993.

- [BOU93b] BOUKIR S., BOUTHEMY P., CHAUMETTE F., JUVIN D. Suivi de segment temps réel dans un contexte de vision active. *2ème Conf. sur l'interface des mondes réels et virtuels, Montpellier, Mars 1993.*
- [BOU93c] BOUKIR S., CHAUMETTE F., BOUTHEMY P., JUVIN D. An active robot vision system for real-time 3D structure recovery. *IFAC Workshop on Intelligent Autonomous Vehicles, Southampton, UK, Avril 1993.*
- [BRA84] BRADY M., YUILLE A. An extremum principle for shape from contour. *IEEE Trans. on PAMI*, Vol. 6, No. 3:pp. 288–301, Mai 1984.
- [BRO92] BROWN C. Issues in selective perception. *11th ICPR Int. Conf. on Patt. Recog.*, Vol. 1:pp. 21–24, Août 1992.
- [CHA89] CHAUMETTE F., RIVES P. Réalisation et calibration d'un système expérimental de vision composé d'une caméra mobile embarquée sur un robot manipulateur. *Rapport de recherche INRIA*, No. 454, Février 1989.
- [CHA90] CHAUMETTE F. La relation vision-commande: théorie et application à des tâches robotiques. *Thèse de l'Université de Rennes 1*, Juillet 1990.
- [CHA92] CHAUMETTE F., BOUKIR S. Structure from motion using an active vision paradigm. *11th IAPR Int. Conf. on Pattern Recognition, La Haye, Pays bas*, Vol 1:pp. 41–44, Septembre 1992.
- [CHE90] CHENG C.L., AGGARWAL J.K. A two-stage hybrid approach to the correspondance problem via forward-searching and backward-correcting. *Proc of 10th ICPR*, Vol. 1:pp. 173–179, Juin 1990.
- [CHI89] CHIEN C.H., AGGARWAL J.K. Model construction and shape recognition from occluding contours. *IEEE Trans. on PAMI*, Vol. 11, No. 4:pp. 372–389, Avril 1989.
- [CRO92] CROWLEY J.L, STELMASZYK P., PUGET P. Measurement and integration of 3-D structures by tracking edge lines. *Int. Journal of Computer Vision*, Vol. 8, No. 1:pp. 29–52, Juillet 1992.
- [CRO93] CROWLEY J.L, BOBET P., MESRABI M. Layered control of a binocular camera head. *Int. Journal of Patt. Recog. and Artif. Intel.*, Vol. 7, No. 1:pp. 109–116, Février 1993.
- [DER87] DERICHE R. Using Canny's criteria to derive an optimal edge detector recursively implanted. *Int. Journal of Computer Vision*, Vol. 1, No. 2:pp. 167–187, Avril 1987.

-
- [DER90] DERICHE R., FAUGERAS O. Tracking line segments. *Image and Vision Computing Journal*, Vol. 8, No. 4:pp. 261–270, Novembre 1990.
- [DHO89] DHOME M., RICHTIN M., LAPRESTE J.T., RIVES G. Determination of the attitude of 3D objects from a single perspective view. *IEEE Trans. on PAMI*, Vol. 11, No. 12:pp. 1265–1278, Décembre 1989.
- [DU 91] DU F., BRADY M., MURRAY D., . Gaze control for a two-eyed robot head. *Proc. of the British Machine Vision Conf., Glasgow*, pp. 193–201, 1991.
- [DUD72] DUDA R.O., HART P.E. Use of the Hough transformation to detect lines and curves in pictures. *Communication of the ACM*, 15, No. 1:pp. 11–15, Janvier 1972.
- [ESP87] ESPIAU B., RIVES P. Closed-loop recursive estimation of 3D features for a mobile vision system. *IEEE Int. Conf. on Robotics and Automation, Raleigh, North Carolina, USA*, Vol. 3:pp. 1436–1443, Avril 1987.
- [ESP92] ESPIAU B., CHAUMETTE F., RIVES P. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, Vol 8, No. 3:pp. 313–326, Juin 1992.
- [ESS88] ESSAFI H. Les processeurs ligne en traitement d’image. *Thèse de l’Université de Toulouse*, No. 307, Juin 1988.
- [FAU87] FAUGERAS O.D., LUSTMAN F., TOSCANI G. Motion and structure from motion from point and line matches. *Proc. 1st Int. Conf. on Computer Vision, Londres*, pp. 25–34, Juin 1987.
- [FAU92] FAUGERAS O.D. What can be seen in three dimensions with an uncalibrated stereo rig. *Proc. 2nd European Conf. on Computer Vision, Santa Margherita Ligure, Italie*, pp. 536–578, Mai 1992.
- [FED89] FEDDEMA J.T, MITCHELL O.R. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 5:pp. 691–700, Octobre 1989.
- [FLA91] FLAMMIA G. Dynamic programming solves approximate planar point matching. *Proc. of 7th SCIA, Aalborg, Danemark*, pp. 266–273, Août 1991.

- [FRA87] FRANKOT R., CHELLAPPA R. A method for enforcing integrability in shape from shading algorithms. *Proc. of 1st Int. Conf. on Computer Vision, Londres*, pp. 118–127, 1987.
- [GUI92] GUISSER L., PAYRISSAT R., CASTAN S. A 3D active vision system with a direct parametrization of surfaces. *Proc. of 7th SCIA Int. Conf. on Image Analysis*, pp. 210–217, 1992.
- [HAR88] HARALICK R.M. Determining camera parameters from the perspective projection of a rectangle. *Intelligent Systems Laboratory, Department of Electrical Engineering, University of Washington*, Février 1988.
- [HER92] HERVE J.Y., ALOIMONOS Y. Exploratory active vision : theory. *Proc. of CVPR 92, Champaign, Illinois*, pp. 10–15, Juin 1992.
- [HOR89a] HORAUD R., CONIO B., LEBoulLEUX O., LACOLLE B. An analytic solution for the perspective 4-point problem. *Publication commune du LIFIA de Grenoble et du TIM3 de Saint-Martin d'Hère*, 1989.
- [HOR89b] HORN B.K.P, BROOKS M.J. *Shape from shading*. MIT Press, Artificial Intelligence, 1989.
- [HUA91] HUANG L., ALOIMONOS J. Relative depth from motion using normal flow : an active and purposive solution. *IEEE Workshop on Visual Motion, New Jersey*, pp. 196–204, Octobre 1991.
- [KEN81] KENDER J. Shape from texture : an agregation transform that maps a class of textures into surface orientation. *Proc. of IJCAI*, pp. 475–480, 1981.
- [LI 91] LI S., MIYAWAKI I., ISHIGURO H., TSUJI S. Realizing active vision by a mobile robot. *IEEE Workshop on Visual Motion, New Jersey*, pp. 205–210, Octobre 1991.
- [LIU86] LIU Y.C., HUANG T.S. Estimation of rigid body motion using straight line correspondences : further results. *Proc. Int. Conf. of Patt.Recog.*, pp. 306–309, Octobre 1986.
- [LON86] LONG-LIMOZIN P., GIRAUDON G. Stereo matching using contextual line region primitives. *Proc. of ICPR 86, Paris*, pp. 974–977, Octobre 1986.

- [LUS87] LUSTMAN F. Vision stéréoscopique et perception du mouvement en vision artificielle. *Thèse de l'Université d'Orsay*, Décembre 1987.
- [MAI85] MAITRE H. Un panorama de la transformée de Hough. *Traitement du Signal*, Vol 2, N. 4, pp. 306–317, 1985.
- [MAR82] MARR D. *-Vision- A computational investigation into the human representation and processing of visual information*. W.H. Freeman and Compagny, 1982.
- [MAR87] MARCÉ L., BOUTHEMY P. Determination of a depth map from an image sequence. *Proc. of the 3rd Int. Conf. on Advanced Robotics ICAR*, pp 221–232, Octobre 1987.
- [MAR93] MARCHAND E. Stratégie de perception d'une scène 3d par vision active. *Rapport de stage IRISA*, 1993.
- [MED84] MEDIONI G., NEVATIA R. Matching images using linear features. *IEEE Trans. on PAMI*, Vol. 6, No. 6:pp. 675–685, Novembre 1984.
- [MIL93] MILIOS E., JENKIN M., TSOTSOS J. Design and performance of trish, a binocular robot head with torsional eye movements. *Int. Jour. of Patt. Recog. and Artif. Intel.*, Vol. 7, No. 1:pp. 51–68, Février 1993.
- [MIT89] MITICHE A., HABELRIH G. Interpretation of straight line correspondences using angular relations. *Pattern Recognition*, Vol. 22, No. 3:pp. 299–308, 1989.
- [MOH91] MOHR R., MORIN L., GROSSO E. Relative positioning with poorly calibrated cameras. *Proc. of the DARPA-ESPRIT workshop on Applications of invariants in Computer Vision, Reykjavik, Islande*, pp. 7–45, Mars 1991.
- [MOR93] MORIN L. Quelques contributions des invariants projectifs à la vision par ordinateur. *Thèse de l'Institut National Polytechnique de Grenoble*, Janvier 1993.
- [MOU89] MOULET D., BARBA D. Méthode de mise en correspondance par relaxation de traitements structuraux pour la segmentation d'une séquence d'images. *AF CET INRIA Reconnaissance des Formes et Intelligence Artificielle*, pp. 1369–1378, Novembre 1989.

-
- [NEG87] NEGAHDARIPOUR S., HORN B.K.P. Direct passive navigation. *IEEE Trans. on PAMI*, Vol. 9, No. 1:pp. 168–176, Janvier 1987.
- [OHT81] OHTA Y., MAENOBU K., SAKAI T. Obtaining surface orientation from texels under perspective projection. *Proc. of 7th IJCAI, Vancouver*, pp. 746–751, 1981.
- [OKU91] OKUTOMI M., KANADE T. A multiple-baseline stereo. *Proc. of CVPR 91, Lahaina, Maui, Hawaii*, pp. 63–69, 1991.
- [PAH92] PAHLAVAN K., EKLUNDH J.O. A head-eye system : Analysis and design. *CVGIP: Image Understanding*, Vol. 56, No. 1:pp. 41–56, Juillet 1992.
- [PEN84] PENTLAND A.P. Local shading analysis. *IEEE Trans. on PAMI*, Vol. 6:pp. 170–187, Mars 1984.
- [POG85] POGGIO T., KOCH C. Ill-posed problems in early vision : from computational theory to analogue networks. *Proc. R. Soc. London B 226*, pp. 303–323, 1985.
- [PRE89] PRESTON K. The Abingdon cross benchmark survey. *IEEE Computer*, Vol 22, No. 7:pp. 9–18, Juillet 1989.
- [RDE86] R. DE SAINT VINCENT A. Perception et modélisation de l'environnement d'un robot mobile: une approche par stéréovision. *Thèse de l'Université Paul Sabatier de Toulouse*, Novembre 1986.
- [RIC91] RICHTIN M., DHOME M., LAPRESTE J.T., RIVES G. Inverse perspective transform using zero-curvature contour points : application to the localization of some generalized cylinders from a single view. *IEEE Trans. on PAMI*, Vol. 13, No. 2:pp. 185–192, Février 1991.
- [RIV89] RIVES P., CHAUMETTE F., ESPIAU B. Visual servoing based on a task-function approach. *1st Int. Symposium on Experimental Robotics, Montréal, Canada*, Juin 1989.
- [SAF92] SAFAEE-RAD R., TCHOUKANOV I., BENHABIB B., SMITH K.C. 3D-pose estimation from a quadratic-curved feature in two perspective views. *11th ICPR Int. Conf. on Patt. Recog., La haye, Pays bas*, Vol. 1:pp. 341–344, Août 1992.

-
- [SAM90] SAMSON C., ESPIAU B. Application of the task function approach to sensor-based-control of robot manipulators. 11th *IFAC World Congress, Tallin, Estonie, URSS*, Vol. 9:pp. 286–291, Août 1990.
- [SAM91] SAMSON C., ESPIAU B., LE BORGNE M. *Robot Control: the Task Function Approach*. Oxford University Press, 1991.
- [SAN86] SANDINI G., TISTARELLI M. Recovery of depth information: camera motion as an integration to stereo. *IEEE Workshop on Motion, Niawah, Islande*, pp. 39–43, Mai 1986.
- [SAN90] SANDINI G., TISTARELLI M. Active tracking strategy for monocular depth inference over multiple frames. *IEEE Trans. on PAMI*, Vol. 12, No. 1:pp. 13–27, Janvier 1990.
- [SCH89] SCHUNK B.G. Image flow segmentation and estimation by constraint line clustering. *IEEE Trans. on PAMI*, Vol. 11, No. 10:pp. 1010–1027, Octobre 1989.
- [SHE87] SHEN J., CASTAN S. Fast binary image correlation algorithm for stereo vision. *Proc. on 5th SCIA, Stockholm*, Vol 1:pp. 317–324, Juin 1987.
- [SHE92] SHEN J., CASTAN S. An optimal linear operator for step edge detection. *CVGIP : Graphical Models and Image Processing*, Vol 54, No. 2:pp. 13–17, Mars 1992.
- [SUB87] SUBBARAO M. Solution and uniqueness of image flow equations for rigid curved surfaces in motion. *Proc. 1st Int. Conf. on Computer Vision, London*, pp. 687–692, Juin 1987.
- [SUB89] SUBBARAO M. Interpretation of image flow: a spatio-temporal approach. *IEEE Trans. on PAMI*, Vol. 11, No. 3:pp. 266–278, Mars 1989.
- [THO93] THOMPSON B., LECHLEIDER P., STUCK E.R. Detecting moving objects using the rigidity constraint. *IEEE Trans. on PAMI*, Vol. 15, No. 2:pp. 162–166, Février 1993.
- [TIS92] TISTARELLI M., SANDINI G. Dynamic aspects in active vision. *CVGIP ; Image Understanding*, Vol. 56, No. 1:pp. 108–129, Juillet 1992.
- [TOS87] TOSCANI G. Système de calibration optique et perception du mouvement en vision artificielle. *Thèse de l'Université d'Orsay*, Décembre 1987.

-
- [TSA86] TSAI R.Y. An efficient and accurate camera calibration technique for 3D machine vision. *Proc. of Int. Conf. on CVPR, Miami Beach*, pp. 364–374, 1986.
- [VAI90] VAILLANT R. Géométrie différentielle et vision par ordinateur : détection et reconstruction des contours d'occultation de surfaces. *Thèse de l'Université d'Orsay*, Décembre 1990.
- [VER90] VERNON D., TISTARELLI M. Using camera motion to estimate range for robotic part manipulation. *IEEE Trans. on Robotics and Automation*, Vol. 6, No. 5:pp. 509–521, Octobre 1990.
- [VIA92] VIALA M. Contribution à la reconstruction de scènes constituées d'objets cylindriques et polyédriques à partir d'une séquence d'images acquises par une caméra en mouvement. *Thèse de l'Université d'Orsay*, Novembre 1992.
- [VIE92] VIEVILLE T., EKHLUND J.-O., PAHLAVAN K., UHLIN T. An example of artificial oculomotor behavior. *7th IEEE Symp. on Intelligent Control, Glasgow*, pp. 348–353, 1992.
- [WAN87] WANG Y.F., MITICHE A., AGGARWAL J.K. Computation of surface orientation and structure of objects using grid coding. *IEEE Trans. on PAMI*, Vol. 9, No. 1:pp. , Janvier 1987.
- [WAN90] WANG X.J., FU J., WU L. A matching algorithm based on hierarchical primitive structure. *Proc. of 10th ICPR, New Jersey*, Vol. 1:pp. 285–287, Juin 1990.
- [WAX87] WAXMAN A.M., PARSİ B.K., SUBBARAO M. Closed-form solutions to image flow equations for 3D structure and motion. *Int. Journal of Computer Vision*, Vol. 1, No. 3:pp. 239–258, Octobre 1987.
- [WEI87] WEISS L.E., SANDERSON A.C. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, Vol. 3, No. 5:pp. 404–417, Octobre 1987.
- [WEL89] WELLS W.M. Visual estimation of 3D lines segments from motion. A mobile robot vision system. *IEEE Trans. on PAMI*, Vol. 5, No 6:pp. 820–825, Décembre 1989.

-
- [WEN89] WENG J., HUANG T.S., AHUJA N. Motion and structure from two perspective views : algorithms, error analysis, and error estimation. *IEEE Trans. on PAMI*, Vol. 11, No. 5:pp. 451–476, Mai 1989.
- [WEN92a] WENG J., COHEN P., REBIHO N. Motion and structure estimation from stereo image sequences. *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 3:pp. 362–382, Juin 1992.
- [WEN92b] WENG J., HUANG T.S., AHUJA N. Motion and structure from line correspondences : closed-form solution, uniqueness, and optimization. *IEEE Trans. on PAMI*, Vol. 14, No. 3:pp. 318–336, Mars 1992.
- [WIT81] WITKIN A.P. Recovering surface shape and orientation from texture. *Artif. Intel.*, Vol. 17:pp. 17–47, Octobre 1981.
- [WU 88] WU J.J., RINK R.E., CAELLI T.M., GOURISHANKAR V.G. Recovery of 3-D location and motion of a rigid object through camera image (an extended Kalman filter approach). *Int. Journal of Computer Vision*, Vol. 3:pp. 373–394, 1988.
- [XIE89] XIE M. Contribution à la vision dynamique : reconstruction d’objets 3D polyédriques par une caméra mobile. *Thèse de l’Université de Rennes 1*, Juin 1989.
- [ZHA90] ZHANG Z. Analyse du mouvement à partir d’une séquence de scènes stéréoscopiques et applications à la robotique mobile. *Thèse de l’université d’Orsay*, Octobre 1990.
- [ZHA92] ZHAO J., SHEN J., CASTAN S. Trinocular stereo vision by window matching in 3D space. *Proc. of 7th SCIA Int. Conf. on Image Analysis*, pp. 202–209, 1992.
- [ZHE91] ZHENG J.Y., CHEN Q., KISHINO F., TSUJI S. Active camera controlling for manipulation. *Proc. of CVPR 91, Lahaina, Maui, Hawaii*, pp. 413–418, Juin 1991.