

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

T H E S E

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice-Sophia Antipolis

Mention: Automatique, Traitement du Signal et des Images

présentée et soutenue par

Vincent BRANDOU

Stéréovision Locale et Reconstruction 3D/4D

Thèse dirigée par Patrick RIVES et Michel PERRIER

soutenue le 9 décembre 2008

Jury:

M. Philippe MARTINET
M. Peter STURM
M. Michel PERRIER
M. Patrick RIVES
M. Ezio Malis
M. Pierre-Marie SARRADIN

Rapporteur
Rapporteur
Examineur
Examineur
Examineur
Invite

Mis en page avec la classe thloria.

Remerciements

Je souhaiterais remercier tout d'abord les membres du jury de cette thèse :

- M. Philippe Martinet pour avoir accepté la présidence du jury et d'avoir également été un des rapporteurs de ce manuscrit ;
- M. Peter Sturm pour le temps passé à l'analyse et à la critique de ce manuscrit en tant que rapporteur ;
- M. Patrick Rives, directeur de cette thèse, dont l'expérience et les précieux conseils m'ont été très utiles tout au long de ces quatre années, et qui a su me motiver dans les moments de découragement avec une grande patience ;
- M. Michel Perrier, qui a co-dirigé cette thèse, et qui a grandement aidé à la réalisation des expérimentations et au bon déroulement de cette thèse avec un suivi régulier de l'état d'avancement des travaux de recherche, merci de m'avoir fait confiance ;
- M. Ezio Malis, co-encadrant de ce travail, chercheur très brillant jamais en panne d'idées, d'une grande aide pour l'élaboration de la méthode et l'analyse des nombreux problèmes rencontrés en expérimentation, et que je remercie également pour ça bonne humeur omniprésente ;
- M. Pierre-Marie Sarradin, membre invité de ce jury, biologiste au sein de l'IFREMER et chef de mission de la campagne en mer MORARETO lors de laquelle j'ai eu l'opportunité de tester l'instrumentation en conditions réelles d'utilisation, et qui a eu la gentillesse d'accepter de faire partie de ce jury.

Ce travail n'aurait pas été possible également sans l'aide d'Anne-Gaëlle ALLAIS présente du début à la fin de ce projet à toutes les étapes, avec qui j'ai partagé son bureau pendant deux ans, et qui m'a toujours encouragé et soutenu à l'approche des deadlines ainsi que pour les expérimentations importantes, un grand merci également pour toutes les fautes corrigées en anglais comme en français.

Je remercie tous les membres de l'ancien projet ICARE et tous les nouveaux membres qui ont rejoints le projet AROBAS de l'INRIA à Sophia-Antipolis, responsables, étudiants et ingénieurs, pour l'aide apportée à mon travail quand j'en avais besoin, mais aussi pour tous les moments partagés ensemble que ce soit aux pauses café, aux parties de poker, ou pour faire du sport. Je n'oublie pas Adan, Mexicain au grand cœur, qui a bien voulu partagé son bureau avec moi et me supporter pendant de nombreux mois.

Un grand merci également à toutes les personnes faisant partie de l'IRFREMÉR qui ont participé de près ou de loin à ce travail, je pense à M. Vincent Rigaud pour m'avoir accueilli dans son département, au service Prao dont j'ai fait partie et où chaque membre participe à créer une très bonne ambiance de travail, au personnel de GENAVIR, à l'équipage du *Pourquoi Pas ?* et à Josée Sarrazin et Pierre-Marie Sarradin pour la campagne en mer aux Açores qui reste pour moi un merveilleux souvenir. Je ne peux pas oublier de remercier évidemment les membres de l'Association des Grosses Feignasses de Janas, qui se reconnaîtront, avec qui j'ai passé de nombreuses heures à courir dans la bonne humeur et dans un cadre magnifique, et comme dirait Guy "elle est pas belle la vie?". Un merci tout spécial à Jean-Pierre Gilliotte, le Mac Giver de l'IFREMÉR, toujours prêt à rendre service avec le sourire, sachant absolument tout fabriquer avec lui rien d'impossible, c'est également un modèle sur le plan sportif, j'aimerais pouvoir faire à 50 ans passés, des courses à pied de 100 km ! Merci à Dominique Santarelli pour les nombreuses discussions que nous avons eu ensemble et de m'avoir donné envie de me mettre à l'équitation, passion qui ne me quitte plus aujourd'hui. Un très grand merci à Cédric, compagnon de galère, avec qui je pouvais décompresser en plaisantant de tout dans les moments les plus durs, et qu'est-ce que ça fait du bien de rire dans les périodes de stress intense.

Je tiens à remercier mes amis qui m'ont soutenu, bien qu'ils aient toujours du mal à savoir ce que je fais exactement comme travail ("c'est quoi ton truc de robot déjà?"), je ne préfère pas mettre de noms au risque d'oublier quelqu'un, mais un grand merci à vous tous.

Pour finir je remercierai les personnes les plus importantes à mes yeux, qui ont contribué par leur soutien à la réussite de ce travail, il s'agit évidemment de ma famille, et en particulier mes parents, ma sœur Frédérique et Sylvain son mari.

Ce manuscrit n'aurait peut être jamais vu le jour sans le soutien durant les derniers mois d'écriture de Marion, que j'aime tendrement.

à mes parents

Table des matières

Introduction

1	Contexte	1
2	Motivation et positionnement du problème	1
2.1	La vision sous-marine	2
2.2	Nature des objets à étudier	3
3	Présentation générale du travail de thèse	3
3.1	Principe d'acquisition des images	3
3.2	Reconstruction 3D métrique dense	4
4	Contribution	5
5	Structure de la thèse	5

Chapitre 1 Etat de l'art et modélisation **7**

1.1	Modélisation géométrique d'un capteur de vision	8
1.1.1	Transformation Scène - Caméra (géométrie projective)	8
1.1.2	Modélisation de la caméra	9
1.1.2.1	La Projection Caméra - Rétine	9
1.1.2.2	La transformation Rétine - Image	10
1.1.2.3	La calibration	11
1.1.3	Géométrie épipolaire	12
1.2	Extraction, appariement et suivi de primitives	15
1.2.1	Extraction de primitives dans les images sous-marines	15
1.2.2	Mise en correspondance	18
1.2.2.1	Appariement des primitives	18
1.2.2.2	Méthodes d'appariement dense	20
1.2.2.3	Contraintes géométriques	23
1.2.3	Suivi visuel	24
1.2.3.1	Méthodes de corrélation	25

1.2.3.2	KLT	25
1.2.3.3	ESM	26
1.3	La reconstruction 3D	27
1.3.1	Reconstruction à partir de la géométrie épipolaire	27
1.3.2	Reconstruction à partir de la texture	30
1.3.3	Reconstruction à partir des silhouettes (aspect)	30
1.3.4	Reconstruction à partir du mouvement	31
1.4	L’asservissement visuel	34
1.4.1	La vision active	34
1.4.2	La vision stéréoscopique	34
1.4.3	Contrôle d’une caméra par fonction de tâche	35
1.4.3.1	Formalisme de la fonction de tâche	35
1.4.3.2	La matrice d’interaction	36
1.4.3.3	Commande dans l’espace 2D	37
1.5	Conclusion	38
Chapitre 2 L’acquisition des images		39
2.1	Principe de la méthode d’acquisition par génération de trajectoires	40
2.1.1	Cas mono-dimensionnel	42
2.1.2	Cas multi-dimensionnel	42
2.2	Simulations du suivi de trajectoire par asservissement visuel	47
2.2.1	Asservissement visuel 2D classique	47
2.2.2	Asservissement visuel invariant aux paramètres intrinsèques des caméras	49
2.2.3	Simulations en présence d’erreurs et de perturbations	55
2.3	Validation et résultats expérimentaux	59
2.3.1	Moyens expérimentaux	59
2.3.2	Validation de l’approche	59
2.3.2.1	Conditions expérimentales optimales	59
2.3.2.2	Méthode générale	59
2.3.2.3	Premiers résultats expérimentaux	60
2.3.3	Modifications du traitement d’images aux objets 3D	63
2.3.3.1	Extraction et appariement	64
2.3.3.2	Suivi dans l’image	64
2.3.4	Contrôle du déplacement du robot	72

2.3.4.1	Impact de l'estimation des paramètres intrinsèques, et des profondeurs des points sur l'asservissement visuel	72
2.3.4.2	Influence de mauvais appariements et de la dérive de points sur le comportement de l'asservissement visuel	72
2.3.4.3	Précision et répétabilité de l'asservissement visuel (commande 2D/invariante)	72
2.3.5	Génération de trajectoires	76
2.3.6	Développement d'un prototype opérationnel	77
2.4	Conclusion	84
Chapitre 3 La reconstruction 3D		85
3.1	Description de la méthode	86
3.1.1	La méthode de stéréophotogrammétrie	86
3.1.2	Reconstruction euclidienne éparse de la structure	86
3.1.2.1	Extraction et appariement de points	88
3.1.2.2	Calcul de la structure 3D éparse par ajustement de faisceaux	88
3.1.3	Reconstruction dense	89
3.1.3.1	Rectification des images	89
3.1.3.2	Appariement dense	90
3.1.3.3	Carte de profondeur dense	91
3.1.4	Rendu du modèle 3D	91
3.1.4.1	Triangulation	91
3.1.4.2	Plaquage de texture	92
3.2	Résultats expérimentaux	93
3.2.1	Modélisation d'objets en 3D	93
3.2.1.1	Conditions expérimentales	93
3.2.1.2	Objets de formes géométriques variées	93
3.2.1.3	Objet cylindrique	99
3.2.1.4	Reconstruction d'un objet sous-marin	110
3.2.2	Evaluation de la précision des mesures sur la structure 3D	116
3.3	Conclusion	118
Conclusion et perspectives		119

Annexe A Logiciel pour l'acquisition d'images stéréoscopiques et pour l'as-	
servissement viuel	123
A.1 Fonctionnalités	123
A.2 Interface graphique	123
A.2.1 Configuration des caméras et affichage	124
A.2.2 Spécifications de la tête stéréo	124
A.2.3 Suivi dans l'image	124
A.2.4 Envoi de la commande au robot	126
Glossaire	131
Bibliographie	135

Table des figures

1.1	Modélisation de la caméra	8
1.2	Le modèle Sténopé	9
1.3	Transformation du repère de la rétine vers le repère image	10
1.4	La géométrie épipolaire	12
1.5	Comparaisons d’algorithmes d’appariement dense sur les images Tsukuba [SSZ01]	20
1.6	Contrainte d’ordonnancement	24
1.7	Géométries affine, projective et euclidienne	29
1.8	Triangulation 2D de Delaunay	29
1.9	Triangulation de Delaunay (bleu) et diagramme de Voronoï (Rouge). C’est le dual de la triangulation de Delaunay. Le polygone centré sur un point p de E est à la frontière de l’ensemble P des points du plan, plus proches de p que tout autre point de E	29
1.10	Méthode de reconstruction voxélique par <i>space carving</i> (illustrations extraites de [KS00])	31
1.11	Reconstruction de l’enveloppe visuelle d’un objet par intersection des cônes d’observation dans l’espace	32
1.12	Reconstruction de la structure d’un objet (en haut à gauche) à partir du champ vectoriel (en haut à droite), permettant de définir une carte de profondeur (en bas à gauche) pouvant ensuite être représentée par un maillage 3D (en bas à droite) [KSK ⁺ 98]	32
2.1	Trajectoires induites par la géométrie de la tête stéréo	40
2.2	Projection d’un point \mathcal{X} de l’espace dans l’image de deux caméras	40
2.3	Description et modélisation des bras manipulateurs	48
2.4	Simulation des trajectoires	49
2.5	Commande 2D avec les caméras calibrées et sans perturbations	50
2.6	Simulation d’une trajectoire par deux asservissements visuels successifs par la commande 2D, avec des paramètres intrinsèques différents entre l’image courante et l’image de référence.	51
2.7	Simulation d’une trajectoire par deux asservissements visuels successifs avec des paramètres intrinsèques différents et la commande invariante.	54
2.8	Projection d’un nuage de points vu par deux caméras stéréo ((a) et (b)) et reprojété dans l’espace invariant ((c) et (d))	55
2.9	Simulation du comportement de la fonction de tâche dans l’espace invariant en présence de bruit sur les coordonnées des points dans l’image	56

2.10	Simulation du comportement de la fonction de tâche dans l'espace invariant lors d'une erreur sur la profondeur d'un point (valeur réelle divisée par 100) sur 100 points suivis	57
2.11	Simulation du comportement de la fonction de tâche dans l'espace invariant lors de la perte d'un point mal apparié sur 100 suivis	58
2.12	Conditions expérimentales	61
2.13	Asservissement visuel dans l'espace invariant	61
2.14	Fonction de tâche du 1 ^{er} asservissement	62
2.15	Fonction de tâche du 2 ^{ième} asservissement	62
2.16	Mise en place des expérimentations dans l'air avec le bras atelier TAO de l'Ifremer	63
2.17	Points SIFT mis en correspondance sur des images sous-marines	65
2.18	Points KLT appariés dans les images stéréo	66
2.19	Rectification des images issues de la tête stéréo	67
2.20	Vérification de la similarité [Bir07]	68
2.21	Résultats de l'asservissement visuel entre deux caméras identiques en utilisant la commande invariante. La géométrie des caméras est de 30cm avec un angle de 15°.	69
2.22	Représentation des points de référence et suivis dans l'image de référence et dans l'espace invariant, après avoir convergé par asservissement visuel.	70
2.23	Commande invariante filtrée appliquée à la caméra gauche	71
2.24	Commande invariante avec 50% d'erreur sur \mathbf{K}	73
2.25	Trajectoire par asservissement visuel avec la commande 2D appliquée au bras TAO. Les caméras transparentes représentent les positions théoriques, celles en vert correspondent à la caméra droite de la tête stéréo, et celles en rouge aux positions de la caméra gauche. La géométrie de la tête stéréo est de 30 cm entre les caméras, et de 10° sur la caméra droite.	78
2.26	Tête stéréo IRIS composée d'une caméra mobile, d'une caméra fixe, de deux projecteurs à LED, et d'une boîte de jonction (absente sur la photo en haut à droite). La structure en aluminium fait 710 mm de largeur, 479 mm de hauteur, et 350 mm de profondeur.	79
2.27	Mise en place des expérimentations dans l'eau avec le bras atelier TAO	80
2.28	Expérimentations avec le ROV <i>victor6000</i>	81
2.29	Acquisition des images servant à la reconstruction sur un site à 1700m de profondeur	82
2.30	Trajectoire théorique pour les essais en mer de la tête stéréo IRIS effectués avec le bras MAESTRO	83
3.1	Méthode générale de reconstruction 3D	87
3.2	Coupure d'un graphe sur une image (3 × 3) avec deux terminaux	90
3.3	Images stéréo d'objets de forme géométrique	94
3.4	Estimation de la structure des objets après triangulation	95
3.5	Estimation de la structure des objets après minimisation par ajustement de faisceaux	96

3.6	Correction par ajustement de faisceaux de l'estimation de la position de la caméra gauche après un asservissement visuel (1 quaternion (1×4) correspondant à la rotation, et un vecteur de translation (1×3))	97
3.7	Images stéréo rectifiées	97
3.8	Carte de disparité dense sur les images rectifiées avec la méthode de Graph-Cut	97
3.9	Structure 3D des objets avec texture	98
3.10	Images stéréo d'un cylindre recouvert d'un poster	99
3.11	Estimation d'une partie de la structure après triangulation	100
3.12	Estimation d'une partie de la structure après minimisation par ajustement de faisceaux	101
3.13	Correction par ajustement de faisceaux de l'estimation des paramètres de la position de la caméra droite après un déplacement par programme robot (1 quaternion (1×4) correspondant à la rotation, et un vecteur de translation (1×3))	102
3.14	Images stéréo rectifiées	102
3.15	Carte de disparité dense sur les images rectifiées avec la méthode de Graph-Cut	103
3.16	Structure 3D des objets avec plaquage de texture	104
3.17	Images stéréo d'un cylindre recouvert de morceaux d'écorce	105
3.18	Estimation d'une partie de la structure de l'objet après triangulation	106
3.19	Estimation d'une partie de la structure de l'objet après minimisation par ajustement de faisceaux	107
3.20	Correction par ajustement de faisceaux de l'estimation des paramètres de la position de la caméra droite après un déplacement par programme robot (1 quaternion (1×4) correspondant à la rotation, et un vecteur de translation (1×3))	107
3.21	Images stéréo rectifiées	107
3.22	Carte de disparité dense sur les images rectifiées avec la méthode de Graph-Cut	108
3.23	Structure 3D de l'écorce avec plaquage de texture	109
3.24	Images stéréo d'une scène sous-marine	110
3.25	Calibration de la tête stéréo IRIS sur le site d'expérimentation	111
3.26	Extraction et appariement des points SIFT dans les images stéréo	112
3.27	Correction des points 3D et de l'estimation des positions des caméras	113
3.28	Correction par ajustement de faisceaux de l'estimation des paramètres de la position de la caméra gauche après un déplacement par programme robot (1 quaternion (1×4) correspondant à la rotation, et un vecteur de translation (1×3))	114
3.29	Images stéréo rectifiées	114
3.30	Carte de disparité dense avec la méthode de Graph-Cut	114
3.31	Reconstruction 3D d'un objet sous-marin à partir d'images stéréo	115
3.32	Mesures métriques sur une structure 3D	116
3.33	Fiabilité des mesures métriques faites sur une reconstruction 3D obtenue à partir d'une paire d'images	117

Table des figures

A.1	Interface graphique du logiciel pour l'asservissement visuel	125
A.2	Configuration de l'affichage des caméras et de l'enregistrement des images .	126
A.3	Paramètres de la tête stéréo	127
A.4	Configuration et visualisation du suivi de points	128
A.5	Visualisation de la commande dans le repère caméra et dans le repère pince avant d'être appliquée sur le bras manipulateur	129

Introduction

1 Contexte

L'océan profond couvre les deux tiers de la surface de notre planète. Les conditions de vie dans ce milieu sont souvent extrêmes. Depuis leur découverte, les écosystèmes profonds tels que les communautés hydrothermales, ont bénéficié d'un effort important d'exploration, de description et de compréhension. Aujourd'hui, l'enjeu scientifique concerne l'étude de leur dynamique spatiale et temporelle afin d'identifier quels sont les facteurs qui contrôlent leur structure et leur fonctionnement. Afin de répondre à ce besoin, nous avons développé une méthode utilisant un système optique, permettant l'acquisition d'images stéréoscopiques sous-marines. Le principe de ce système est basé sur les techniques de mesure par photogrammétrie et consiste à déterminer les coordonnées en trois dimensions d'un point d'un objet à partir de ses projections sur plusieurs images photographiques prises à partir de différentes positions. L'objectif est de réaliser la reconstruction tridimensionnelle de structures sous-marines naturelles, rigides et statiques, de petite échelle (de l'ordre du mètre cube), ceci à des fins de mesures quantitatives. Un des moyens dont disposent les chercheurs en biologie marine de l'IFREMER (Institut français de recherche pour l'exploitation de la mer) pour étudier les fonds océaniques est un robot sous-marin, le ROV (Remote Operated Vehicle) *victor6000*, pouvant atteindre une profondeur de 6000 mètres. C'est à partir de ces moyens techniques, un robot muni d'un bras manipulateur, et un système de caméras stéréo, que nous avons développé notre méthode de reconstruction.

2 Motivation et positionnement du problème

La reconstruction 3D à partir d'une séquence d'images a déjà fait l'objet de très nombreuses recherches depuis plusieurs années, étant données l'étendue des domaines d'application et l'évolution importante des moyens de calcul. Les méthodes existantes diffèrent beaucoup suivant le type d'objet à reconstruire (forme, dimension, texture, ...) et les connaissances a priori du système ou de l'environnement (paramètres intrinsèques et extrinsèques des caméras, profondeurs, éclairage, scène structurée, ...). L'aspect innovant de nos recherches sur ce thème se situe au niveau de l'approche pour l'acquisition des images servant au calcul du modèle 3D de la scène, répondant aux contraintes du milieu associées aux fonds océaniques.

L'objectif applicatif de cette thèse est de proposer une méthodologie complète de reconstruction 3D d'objets sous-marins à des fins de mesures quantitatives. Ceci afin que les

biologistes marins puissent étudier l'évolution de l'écosystème dans les zones profondes au cours du temps.

Les différents enjeux portent donc sur l'acquisition et l'exploitation des images de scènes naturelles complexes, à travers l'étude des différentes possibilités d'extraction d'informations, de l'appariement des points de façon robuste ainsi que de leur suivi dans la séquence d'images. L'enjeu final porte sur une reconstruction 3D partielle d'une scène ou d'un objet. Une approche originale regroupant l'acquisition des images et la reconstruction, a été développée afin d'être utilisable pour réaliser des expérimentations en conditions réelles.

2.1 La vision sous-marine

L'exploitation d'images sous-marines est rendue fortement difficile par les contraintes liées au milieu océanique. En effet l'eau de mer introduit des effets variables sur la propagation des signaux dégradant la qualité de l'image. Les principales causes de cette dégradation sont dues à la présence de particules, à l'absorption de l'énergie des rayons lumineux ainsi qu'à des problèmes d'éclairage.

Voici un certain nombre de difficultés spécifiques au milieu sous-marin :

- L'intensité de la lumière qui pénètre dans les océans diminue rapidement avec la profondeur, l'utilisation de sources de lumière artificielle devient donc indispensable pour observer la scène avec des caméras CCD. Cette source de lumière n'étant présente que sur l'engin sous-marin, l'éclairage ne peut pas être homogène, provoquant ainsi des zones d'ombre dans l'image. De plus la puissance d'éclairage doit être adaptée suivant le coefficient d'absorption de l'eau, et suivant la couleur plus ou moins claire du fond reflétant la lumière.
- La présence de particules en suspension provoque des phénomènes de réfraction et de réflexion des rayons lumineux. La réfraction représente la déviation que subit le rayon lumineux en passant d'un milieu à un autre avec des indices de réfraction différents. Le déplacement du robot sous-marin à l'aide d'hélices à proximité du fond, augmente souvent de façon importante la turbidité de ces particules suivant la nature du sol.
- L'indice optique de l'eau est un paramètre variable en fonction des caractéristiques du milieu qui sont la température, la salinité et la pression de l'eau. Cette variation influe directement sur les paramètres intrinsèques des caméras, nécessaires pour une reconstruction euclidienne.
- Les hublots des enceintes utilisées pour protéger les caméras de l'eau et de la pression provoquent généralement des phénomènes de distorsion optique. Cependant il existe des hublots pouvant réduire jusqu'à 99% ce phénomène de distorsion.
- L'environnement naturel dans lequel évolue l'engin sous-marin ne peut pas être contrôlé, ainsi il est impossible d'éviter des occultations dues à des objets en mou-

vement tels que des poissons, ou à la remontée de fluides hydrothermaux (riche en gaz dissous et en métaux).

2.2 Nature des objets à étudier

Un exemple type d'objet sous-marin à modéliser est les cheminées hydrothermales¹ présentes au niveau des dorsales océaniques. Mis à part les fluides sortant de ses cheminées, l'objet est considéré immobile, rigide et opaque. Nous limitons le volume des objets à reconstruire à environ un mètre cube au maximum, à cause de la portée limitée de l'éclairage des projecteurs et afin de limiter les déplacements du robot dont le positionnement au sol est souvent difficile, en fonction des obstacles et de l'état du sol.

3 Présentation générale du travail de thèse

Toute méthode de reconstruction métrique utilisant différentes vues d'une caméra, nécessite la détermination des paramètres intrinsèques et des positions de la caméra correspondant à l'acquisition des différentes vues, si ceux-ci ne sont pas déjà connus. A partir de l'estimation de ces paramètres, il peut être facile de retrouver par triangulation les coordonnées 3D d'un ensemble de points extraits et mis en correspondance dans les images, appartenant à la structure de l'objet.

Les différents paramètres d'une caméra sont estimés par des méthodes d'une précision relative. De plus, les conditions sous-marines d'exploitation remettent souvent en cause la validité des modèles et des méthodes de calibrage. Il est donc nécessaire d'utiliser un algorithme de minimisation lors du calcul de reconstruction, afin de converger vers les paramètres réels de la caméra et ainsi obtenir une reconstruction fidèle de l'objet. Ainsi, une première constatation que nous formulons, est que la précision finale de la reconstruction 3D d'un objet est fortement conditionnée par l'estimation des paramètres extrinsèques et intrinsèques de la caméra utilisée.

Sachant que l'on désire une reconstruction 3D métrique la plus précise possible pour réaliser des mesures, nous avons choisi de nous intéresser plus précisément à l'estimation de ces paramètres. La détermination des paramètres intrinsèques passe par une phase de calibration avec une mire par exemple, ou d'autocalibration nécessitant l'application de mouvements spécifiques à la caméra. L'estimation des paramètres extrinsèques est un problème difficile étant donné nos conditions expérimentales dans les fonds océaniques. C'est ce problème qui a conduit à l'élaboration d'un procédé original d'acquisition d'images qui représente la contribution essentielle de ce travail de thèse.

3.1 Principe d'acquisition des images

Le moyen le plus simple pour connaître les différentes positions de la caméra est de contrôler son déplacement, sur une trajectoire imposée et connue. Si de plus le déplacement entre les positions successives de la caméra est le même, c'est-à-dire même translation

¹Évent situé à proximité des dorsales océaniques, qui évacue une partie de la chaleur interne de la Terre.

et même rotation, on peut alors exprimer chaque position de la caméra en fonction du déplacement imposé et de la première position de la caméra. Ainsi on réduit le nombre d'inconnues intervenant dans le calcul de la reconstruction, et cela permet de couvrir toute la surface d'un objet par des prises de vues à intervalles réguliers. On considèrera que la première position de la caméra est associée au repère monde, les autres positions de la caméra seront exprimées par rapport à la première position. Ceci nous amène à formuler une seconde constatation : si chaque position de la caméra est obtenue après un même déplacement dans l'espace, alors on peut diminuer de manière significative le temps de calcul de la reconstruction 3D, en diminuant le temps de recherche des paramètres de la caméra.

Pour arriver à déplacer une caméra sur une position désirée, la solution envisagée est de réaliser un asservissement. Pour ce faire, l'utilisation d'un bras manipulateur avec un robot stable posé au sol semble être la meilleure solution. Ainsi on limite les perturbations sur le mouvement de la caméra dues aux courants marins et à la stabilisation du robot. En contrepartie l'espace de travail du bras manipulateur sera limité par les butées articulaires et la longueur des axes.

Pour réaliser le contrôle du déplacement du manipulateur, deux solutions sont envisageables : soit utiliser une approche en boucle ouverte sur l'environnement, par exemple, un asservissement en position du bras, soit utiliser une approche en boucle fermée, par exemple, une technique d'asservissement visuel. Sachant que nous allons évoluer dans un environnement sous-marin naturel, l'accent est donc mis sur la flexibilité du système. Or, utiliser le bras de l'engin sous-marin en asservissement en position, revient à programmer à l'avance toutes les positions nécessaires de l'organe effecteur, correspondantes chacune à une prise de vue. Le nombre de positions à programmer devient vite important en fonction de la taille et de la forme de l'objet, et si l'on prend en considération l'inclinaison entre le sol sur lequel sera posé l'engin et l'objet à reconstruire.

Le choix que nous allons argumenter dans la suite de ce manuscrit, s'est porté sur une approche par asservissement visuel nous permettant de nous affranchir de la génération de trajectoire, et de réaliser un contrôle précis du positionnement de la caméra vis-à-vis de la scène à reconstruire. L'avantage d'un tel système est également de pouvoir l'utiliser sur n'importe quel bras manipulateur sans avoir à faire de modification, si ce n'est prendre en compte l'espace de travail du bras. Nous présenterons dans ce travail, un système actif d'acquisition automatique d'images basé sur une méthode d'asservissement visuel, et permettant de garantir un positionnement précis des prises de vues suivant une géométrie prédéterminée.

3.2 Reconstruction 3D métrique dense

La reconstruction 3D est un domaine très actif depuis maintenant de nombreuses années, dont les applications sont aussi nombreuses que les méthodes développées. Ainsi il existe des méthodes spécifiques pour modéliser des bâtiments, des arbres, des organes, ou des visages. Avant de vouloir modéliser un objet, il faut donc d'abord s'intéresser à sa nature, à ses propriétés. Le fait que l'originalité de ce travail soit principalement la méthode d'acquisition, plutôt que la méthode de reconstruction implémentée dans cette thèse, vient du fait que ce sont les conditions expérimentales liées au milieu océanique qui

sont véritablement nouvelles, plus que les objets sous-marins en eux-mêmes, qui sont des objets naturels sans modèle de forme ou de texture bien défini. Sur la partie reconstruction 3D, nous nous sommes donc surtout attachés à fournir une méthode avant tout robuste aux perturbations du milieu sous-marin qui se répercutent dans les images, la méthode restant générique pour tout type d'objet.

L'une des grandes difficultés dans la construction d'un modèle 3D d'une scène ou d'un objet, est la mise en correspondance des primitives extraites dans la séquence d'images. Ces primitives peuvent être d'ordre métrique, analytique, structurel ou symbolique, et doivent être particulièrement robustes à de nombreuses perturbations ou transformations dans l'image, comme le changement d'échelle, d'éclairage ou le bruit dans l'image. La fiabilité de la mise en correspondance va dépendre directement du niveau de précision des informations caractérisant les primitives, une erreur d'appariement entraînant automatiquement une erreur sur l'estimation d'un point 3D. En règle générale, plus les critères de sélection sont sévères, plus le nombre de primitives à apparier sera restreint. Or, une reconstruction métrique dense est nécessaire pour permettre des mesures précises de dimensionnement, ce qui implique d'avoir un très grand nombre d'appariements entre les images, on parle alors d'appariement dense (*dense matching*). Il est donc nécessaire de procéder en plusieurs étapes que nous détaillerons dans la suite de ce manuscrit, dont la première est l'estimation approximative de la structure du modèle 3D et de la géométrie à partir d'un nuage de points épars. La seconde partie est consacrée à l'appariement dense qui est l'étape la plus lourde en terme de temps calcul. Enfin la dernière partie présente le rendu final, avec la simplification du modèle par un maillage, dont l'aspect réaliste sera renforcé par un plaquage de texture.

4 Contribution

L'aspect innovant de cette thèse porte essentiellement sur l'approche développée pour faciliter la reconstruction 3D, à travers l'acquisition d'images à intervalles réguliers le long d'une trajectoire prédéfinie. L'enjeu est d'avoir une reconstruction plus précise en contraignant le mouvement des caméras par notre méthode d'acquisition, par rapport à un système non guidé (*free-hand*). Un travail important a été d'étudier et de choisir parmi de nombreuses méthodes de traitement d'images existantes, les plus appropriées à nos contraintes expérimentales, et répondant aux résultats escomptés. Une autre grande partie du travail effectué a porté sur la réalisation d'un prototype opérationnel de tête stéréo dans le cadre du projet européen EXOCET/D (EXtreme ecosystem studies in the deep OCEan : Technological Developments), dimensionné pour nos recherches, puis sur sa mise en œuvre lors des nombreuses expérimentations menées afin de tester et valider nos différents choix méthodologiques et matériels, ainsi que notre approche générale.

5 Structure de la thèse

Ce manuscrit est composé de trois grands chapitres : l'état de l'art, la méthodologie et les expérimentations. Dans le première chapitre nous exposons tout d'abord certains

concepts de base en vision par ordinateur afin de mieux appréhender les différentes méthodes exposées par la suite. Nous présentons ensuite un état de l'art des méthodes de reconstruction 3D pouvant être envisagées sur des images sous-marines. L'approche originale développée pour faire la reconstruction 3D s'appuie sur une technique d'asservissement visuel, ce qui nous a amenés à faire également un état de l'art des méthodes existantes. Enfin nous nous sommes intéressés aux méthodes robustes d'extraction et d'appariement de primitives, le plus souvent indispensables pour faire en traitement d'image du suivi ou de la reconstruction 3D.

Le second chapitre développe la méthode d'acquisition des images servant à la reconstruction 3D, à travers l'asservissement visuel utilisé et les différentes expérimentations qui ont été menées.

Le troisième et dernier chapitre est centré sur la reconstruction 3D, avec le détail de la méthode utilisée et les résultats obtenus.

Enfin nous terminerons ce manuscrit par un bilan faisant à la fois la synthèse du travail effectué, et ouvrant des perspectives de recherche applicative sur les approches développées.

Chapitre 1

Etat de l'art et modélisation

Ce premier chapitre permet de présenter un état de l'art des différentes méthodes intervenant dans chacune des étapes de notre méthode globale de reconstruction 3D, après avoir posé les bases de quelques concepts géométriques en vision par ordinateur. Notre méthode de reconstruction 3D s'appuie sur une technique d'asservissement visuel. C'est pour cette raison que les premières méthodes présentées concernent l'extraction, l'appariement, et le suivi de primitives, à la base de nombreuses méthodes d'asservissement visuel et de reconstruction 3D exposées également à la fin de ce chapitre.

caméra. Dans les coordonnées homogènes, les translations, rotations, transformées perspectives sont représentées par des produits de matrices. Ainsi la transformation entre le repère caméra et le repère monde pour un point $\mathbf{P}_M = (x_M, y_M, z_M, 1)^\top$ de l'espace s'écrit :

$$\mathbf{P}_C = \mathbf{T}_M^C \mathbf{P}_M \quad (1.1)$$

où le point \mathbf{P}_C de coordonnées $(x_C, y_C, z_C, 1)^\top$ est exprimé dans le repère caméra. La matrice \mathbf{T}_M^C de transformation entre les deux repères a la forme :

$$\mathbf{T}_M^C = \begin{pmatrix} \mathbf{R}_M^C & \mathbf{t}_M^C \\ \mathbf{0} & 1 \end{pmatrix} \quad (1.2)$$

où le vecteur \mathbf{t}_M^C de dimension (3×1) représente la position du repère caméra dans le repère monde. La matrice \mathbf{R}_M^C de dimension (3×3) désigne l'orientation du repère monde dans le repère caméra.

1.1.2 Modélisation de la caméra

1.1.2.1 La Projection Caméra - Rétine

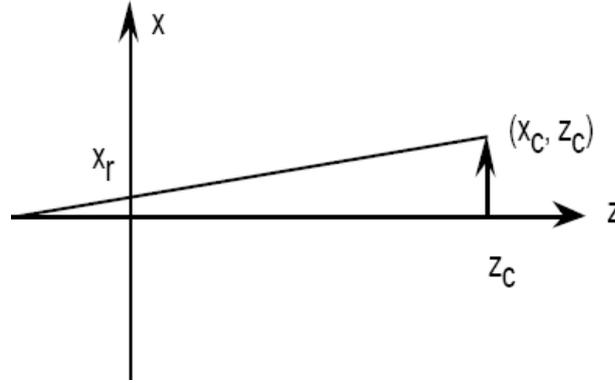


FIG. 1.2 – Le modèle Sténopé

La projection de la caméra vers la rétine est une projection perspective. De nombreux algorithmes de vision s'appuient sur le modèle sténopé pour modéliser la formation de l'image. Ce modèle, également appelé modèle "pin-hole" dans la littérature anglo-saxonne, permet de décrire le processus de formation d'une image par projection des points 3D sur le plan rétinien en des points 2D (Fig. 1.2). Celui-ci suppose que le système optique de la caméra, c'est-à-dire sa lentille, respecte les conditions de Gauss. Ces conditions précisent que l'objet doit être situé au voisinage de l'axe optique et qu'il doit être de petite taille. Soit $\mathbf{P}_C = (x_C, y_C, z_C, 1)^\top$ un point de l'espace exprimé dans le repère caméra se projetant en $\mathbf{P}_R = (x_R, y_R, 1)$ dans le plan de la rétine.

Par triangles semblables on peut écrire :

$$\frac{x_R}{f} = \frac{x_C}{z_C} \Leftrightarrow x_R z_C = f x_C \Leftrightarrow \alpha x_R \frac{z_C}{\alpha} = f x_C \quad (1.3)$$

$$\frac{y_R}{f} = \frac{y_C}{z_C} \Leftrightarrow y_R z_C = f y_C \Leftrightarrow \alpha y_R \frac{z_C}{\alpha} = f y_C \quad (1.4)$$

où f représente la distance focale. On choisit α tel que $\frac{z_C}{\alpha} = 1 \Leftrightarrow z_C = \alpha$.
En rotation matricielle, on obtient :

$$\alpha \begin{pmatrix} x_R \\ y_R \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_C \\ y_C \\ z_C \\ 1 \end{pmatrix} \quad (1.5)$$

où $\begin{pmatrix} \alpha x_R \\ \alpha y_R \\ \alpha \end{pmatrix}$ représente le vecteur de projection.

1.1.2.2 La transformation Rétine - Image

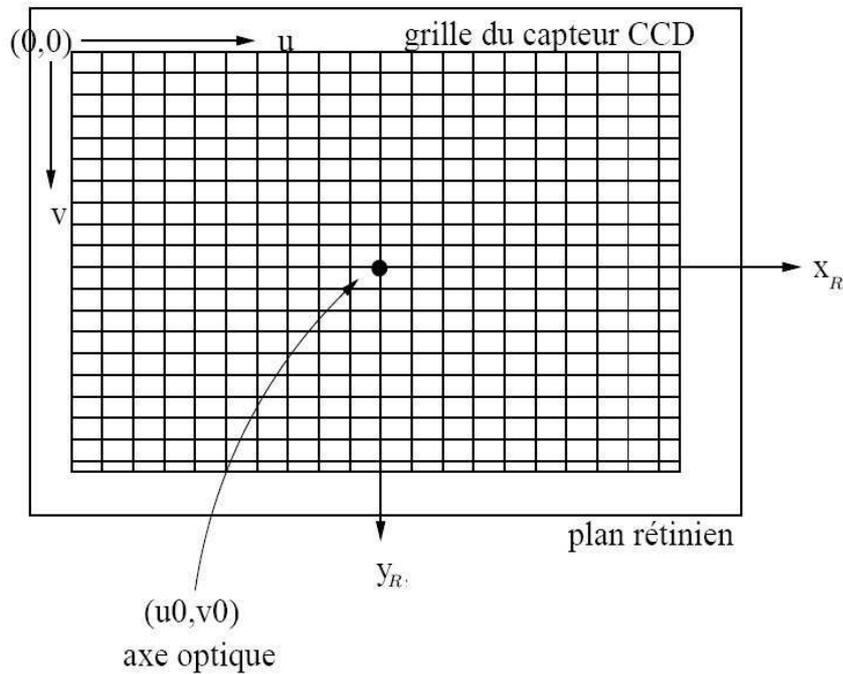


FIG. 1.3 – Transformation du repère de la rétine vers le repère image

Une caméra capture la lumière d'une scène à travers un capteur qui, à partir de la lentille, convertit l'énergie lumineuse en information pixélique. Ainsi, une position dans l'espace est convertie en pixel (picture element) sur une image (Fig. 1.3). Cette conversion

dépend des paramètres intrinsèques de la caméra. Ces paramètres sont :

- f : distance focale (mm),
- (u_0, v_0) : coordonnées de la projection du centre optique de la caméra sur le plan image (en pixels),
- k_u, k_v : facteurs d'agrandissement de l'image (pixel/mm),
- γ : déviation, traduit la non-orthogonalité des cellules électroniques photosensibles qui composent le capteur de la caméra. La plupart du temps, ce paramètre est négligé et prend donc une valeur nulle.

Ainsi ces paramètres peuvent être représentés par une matrice $\mathbf{K}(3 \times 3)$, appelée matrice des paramètres intrinsèques dépendant uniquement de la caméra. Cette matrice décrit le repère du plan rétinale. On peut l'écrire sous cette forme :

$$\mathbf{K} = \begin{pmatrix} k_u & \gamma & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \alpha_u & \alpha_{uv} & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.6)$$

Ainsi la projection d'un point de l'espace \mathbf{P}_M en un point $\mathbf{P}_I = (u, v, 1)^\top$ dans le repère image est égal à :

$$\alpha \mathbf{P}_I = \mathbf{K} \mathbf{P}_C = \mathbf{K} \mathbf{T}_M^C \mathbf{P}_M = \mathcal{P} \mathbf{P}_M \quad (1.7)$$

où \mathcal{P} est la matrice de projection entre le repère monde et le repère image.

$$\mathcal{P} = \mathbf{K} \mathbf{T}_M^C \quad (1.8)$$

1.1.2.3 La calibration

Dès lors qu'une interprétation métrique des images doit être obtenue, la calibration de la caméra est une tâche fondamentale afin de passer d'une reconstruction 3D projective à une reconstruction 3D métrique. Cette tâche consiste à identifier les différents éléments de la matrice des paramètres intrinsèques \mathbf{K} . Ce problème a fait l'objet de nombreux travaux [FT86, LT88, Zha00] qui ont débouché sur des solutions simples et efficaces avec l'utilisation de mires 3D ou planes.

Des méthodes d'auto-calibrage sont progressivement apparues [FLM92, PG97] pour apporter la souplesse d'utilisation faisant défaut au calibrage hors ligne. Le problème de l'auto-calibrage est d'obtenir ces paramètres sans utiliser de mire précisément connue. L'idée principale est donc de remplacer la mire par n'importe quelle scène 3D. La seule connaissance a priori sur la scène est celle de sa rigidité, mais en ajoutant éventuellement d'autres hypothèses sur la caméra, il est alors possible de formuler un ensemble de

contraintes géométriques à partir d'une séquence d'images. Leur mise en application est souvent difficile de part leur instabilité, ces algorithmes étant non-linéaires par essence. Ainsi, certains mouvements de la caméra peuvent conduire à une dégénérescence des contraintes [Stu97a]. Dans tous les cas, la résolution des équations mises en œuvre passe par des algorithmes d'optimisation numérique (minimisation d'une fonction de coût en l'occurrence).

Les différentes méthodes existantes ne seront pas détaillées dans cette thèse, le problème de la calibration ne faisant pas partie de mes travaux de recherche. Cependant il est important de noter que les caméras seront supposées calibrées lors du processus de reconstruction 3D.

1.1.3 Géométrie épipolaire

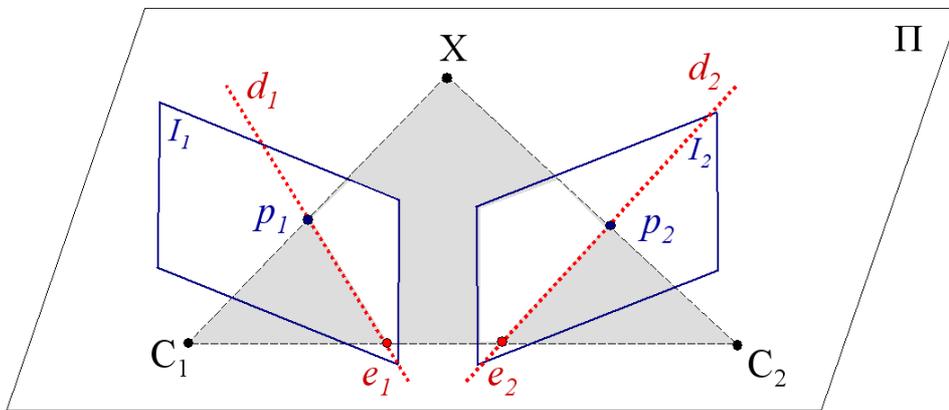


FIG. 1.4 – La géométrie épipolaire

Nous nous intéressons plus spécialement au contexte de la vision stéréoscopique, appellation qui recouvre l'acquisition simultanée de deux images d'une même scène sous deux points de vues différents. Il existe dans ce cas une contrainte forte liant les deux projections d'un point de l'espace dans les deux images.

La géométrie épipolaire décrit les contraintes existantes entre les points \mathbf{p}_1 et \mathbf{p}_2 lorsqu'ils sont les images (projectives) du même point \mathcal{X} sur les deux caméras (Fig. 1.4).

Définition 1 : Le plan épipolaire Π associé au point \mathcal{X} est le plan défini par les trois points \mathcal{X} , C_1 et C_2 .

Définition 2 : Pour un point \mathcal{X} , les intersections du plan $\Pi(\mathcal{X})$ avec les plans images I_1 et I_2 définissent respectivement les droites épipolaires de paramètres \mathbf{d}_1 et \mathbf{d}_2 .

Ainsi, la projection de la droite $(C_1\mathcal{X})$ sur l'image I_2 représente la droite épipolaire \mathbf{d}_2 , correspondant à l'ensemble des points pouvant être associés à \mathbf{p}_1 .

Le point \mathbf{p}_1 appartient à la droite définie par \mathbf{d}_1 si :

$$\mathbf{p}_1^\top \mathbf{d}_1 = 0 \quad (1.9)$$

D'autre part :

$$\mathbf{d}_2 = \mathbf{F}\mathbf{p}_1 \quad (1.10)$$

$$\mathbf{d}_1 = \mathbf{F}^\top \mathbf{d}_2 \quad (1.11)$$

La matrice \mathbf{F} est appelée matrice fondamentale. Elle décrit la géométrie épipolaire du système stéréoscopique. La seule donnée nécessaire pour le calcul de la matrice fondamentale est un ensemble de points appariés (au moins huit points mis en correspondance sont nécessaires). L'algorithme original de la méthode des 8 points est dû à Longuet-Higgins [LH81]. En pratique, plus de huit points appariés étant généralement disponibles, la matrice fondamentale est déterminée par optimisation au sens des moindres carrés. Des techniques d'optimisation linéaire, telle que la méthode de décomposition en valeurs singulières [GR70, GV96, TB97], facile à mettre en oeuvre et rapide en temps de calcul, sont utilisées. Mais les méthodes linéaires permettent surtout d'initialiser une optimisation non linéaire, telle que l'algorithme de Levenberg-Marquart [Lev44, Mar63, GM78], afin d'améliorer la précision du calcul.

La matrice fondamentale "transfère" un point \mathbf{p}_1 de l'image I_1 dans la seconde image I_2 via la relation :

$$\mathbf{p}_2^\top \mathbf{F} \mathbf{p}_1 = 0 \quad (1.12)$$

\mathbf{F} dépend des paramètres intrinsèques ($\mathbf{K}_1, \mathbf{K}_2$) et extrinsèques des caméras suivant :

$$\mathbf{F} = \mathbf{K}_2^\top ([\mathbf{t}]_\times \mathbf{R}) \mathbf{K}_1^{-1} \quad (1.13)$$

Définition 3 : Les épipoles \mathbf{e}_1 et \mathbf{e}_2 sont les projections du centre optique d'une caméra sur le plan image de la seconde.

On notera :

$$\mathbf{e}_2^\top \mathbf{F} = 0, \quad \mathbf{F} \mathbf{e}_1 = 0 \quad (1.14)$$

Théorème : Deux points \mathbf{p}_1 et \mathbf{p}_2 appartenant respectivement aux images I_1 et I_2 , correspondent à un point unique de l'espace si et seulement si \mathbf{p}_1 est sur la ligne épipolaire $\mathbf{d}_1 = \mathbf{F}^\top \mathbf{p}_2$, ou par équivalence, si \mathbf{p}_2 est sur la ligne épipolaire $\mathbf{d}_2 = \mathbf{F}\mathbf{p}_1$.

Lorsque les paramètres intrinsèques des caméras sont connus, il est possible d'exprimer la contrainte épipolaire dans les repères rétiniens. C'est-à-dire, pour un point image \mathbf{p}_1 de l'image I_1 , d'exprimer la contrainte correspondant à :

$$\mathbf{x}_1 = \mathbf{K}_1^{-1} \mathbf{p}_1 \quad (1.15)$$

où \mathbf{K}_1 est la matrice des paramètres intrinsèques de l'image I_1 . Par simple analogie avec ce qui a été vu précédemment, la contrainte épipolaire devient :

$$\mathbf{x}_2^\top \mathbf{K}_2^\top \mathbf{F} \mathbf{K}_1 \mathbf{x}_1 = 0, \quad (1.16)$$

soit :

$$\mathbf{x}_2^\top \mathbf{E} \mathbf{x}_1 = 0 \quad (1.17)$$

avec :

$$\mathbf{E} = \mathbf{K}_2^\top \mathbf{F} \mathbf{K}_1 \quad (1.18)$$

La matrice \mathbf{E} de l'expression précédente est appelée matrice essentielle. Cette matrice est de rang deux, de plus les deux valeurs propres non nulles de \mathbf{E} ont même valeurs. La matrice essentielle peut être estimée, théoriquement, à partir de cinq correspondances.

Comprendre la formation des images d'une caméra est essentiel pour comprendre le schéma inverse exposé par la suite, correspondant à la reconstruction 3D à partir des images 2D. Ces principes de base et ces définitions seront réutilisés tout au long de ce manuscrit, que ce soit pour la reconstruction 3D ou l'asservissement visuel.

1.2 Extraction, appariement et suivi de primitives

Les processus d'extraction et de mise en correspondance (ou appariement) de primitives géométriques entre deux images sont des tâches de bas niveau essentielles en vision par ordinateur, comme en vision dynamique ou en asservissement visuel. Les primitives permettent de suivre des déplacements dans l'image par des méthodes de suivi visuel temps réel afin de pouvoir réaliser un asservissement visuel par le calcul d'une commande.

1.2.1 Extraction de primitives dans les images sous-marines

La première étape que ce soit pour faire de l'asservissement visuel ou de la reconstruction 3D, est d'extraire des informations de l'image sous forme de primitives. Le choix de ces éléments est crucial pour les mettre en correspondance entre les différentes images. Les propriétés intrinsèques caractérisant la primitive doivent permettre une mesure de ressemblance fortement discriminante.

Les méthodes d'appariement proposées dans la littérature se distinguent d'abord selon le type de primitives utilisées :

- **Points** : pixels, points d'intérêt, éléments de contour, points caractéristiques le long d'un contour, jonctions, sont des éléments particulièrement adaptés à la vision stéréo malgré la pauvreté de leur contenu sémantique. La difficulté est d'avoir des caractéristiques permettant une mesure de similarité qui ne soit pas perturbée par la distorsion projective. Lorsque les images sont texturées mais peu structurées, seules des primitives de type point sont susceptibles d'être détectées.
- **Segments** : segments de droite, arcs de cercle, portions de conique,... Ces primitives sont souvent utilisées dans des scènes polyédriques.
- **Régions** : souvent mal adaptées à la vision stéréo, car la distorsion projective modifie la taille et la forme d'une même région sur les images stéréo.

Comparaison des détecteurs classiques de points

Nous avons fait le choix de nous intéresser et donc d'insister sur les primitives sous forme de point, étant donné le fait que nous ne disposons pas de scènes structurées, et que ce genre de primitive est bien adapté aux images stéréo. Les détecteurs de points d'intérêt ont pour but de localiser des points caractéristiques stables dans une image. Un point d'intérêt peut être défini lorsque les valeurs de l'intensité varient beaucoup dans toutes les directions. De nombreuses méthodes existent déjà, mais nous choisissons ici de n'en présenter que certaines parmi les plus connues :

- **Opérateur de Beudet**
Beudet a été le premier à proposer un détecteur de points d'intérêt [Bea78]. Cet opérateur utilise les dérivées secondes du signal. Le détecteur de Beudet est basé

sur l'extraction des maxima locaux de l'opérateur suivant :

$$K(x, y) = I_{xx}I_{yy} - I_{xy}^2 \quad (1.19)$$

où :

$$I_x = \frac{\partial I(x, y)}{\partial x}, \quad I_y = \frac{\partial I(x, y)}{\partial y} \quad (1.20)$$

$$I_{xx} = \frac{\partial^2 I(x, y)}{\partial x^2}, \quad I_{yy} = \frac{\partial^2 I(x, y)}{\partial y^2}, \quad I_{xy} = \frac{\partial^2 I(x, y)}{\partial x \partial y} \quad (1.21)$$

Cet opérateur est invariant à la rotation.

– **Détecteur de coins de Moravec**

L'idée du détecteur de Moravec [Mor77] est de considérer le voisinage d'un pixel (une fenêtre) et de déterminer les changements moyens de l'intensité dans le voisinage considéré lorsque la fenêtre se déplace dans diverses directions. Plus précisément on considère la fonction :

$$E_{dx,dy}(x, y) = \sum_{x,y} w(x, y) |I(x + dx, y + dy) - I(x, y)|^2, \quad (1.22)$$

où $E(x, y)$ représente la moyenne du changement d'intensité lorsque la fenêtre est déplacée de (dx, dy) , $I(x, y)$ est l'intensité du pixel (x, y) de l'image, et w correspond à la fenêtre de voisinage autour du pixel (x, y) (de valeur 1 à l'intérieur de la fenêtre, et nul à l'extérieur).

Cette fonction mesure les différences entre une fenêtre du signal et ses quatre fenêtres voisines, dans les directions parallèles aux lignes et colonnes de l'image. Si la zone image considérée contient un coin ou un point isolé, la fonction E prendra de fortes valeurs dans toutes les directions. Le principe du détecteur de Moravec est donc de rechercher les maxima locaux de E , en chaque pixel au dessus d'un certain seuil indiquant ainsi la présence d'un point d'intérêt.

– **Détecteur de Kitchen et Rosenfeld**

L'approche de Kitchen et Rosenfeld [KR82] repose sur la courbure de courbes planes extraites dans l'image. Les points caractéristiques sont définis par les maxima de courbure des isophotes du signal (ensemble de points de même intensité lumineuse définissant une ligne ou une surface). Les dérivées de l'image sont obtenues par approximation de la surface par un polynôme. L'opérateur utilisé est donné par :

$$K(x, y) = \frac{I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_xI_y}{I_x^2 + I_y^2} \quad (1.23)$$

– **Détecteur de Harris et Stephens**

Harris et Stephens [HS88] ont identifié certaines limitations au détecteur de Moravec. Ils ont alors amélioré l’approche en calculant une matrice liée à la fonction d’auto-corrélation, qui prend en compte les valeurs des dérivées premières du signal sur une fenêtre. Ceci est une amélioration par rapport à Moravec, car la discrétisation utilisée pour calculer la fonction d’auto-corrélation, due au déplacement et aux directions choisies, n’est plus nécessaire. Par voie de conséquence, Harris et Stephen proposent l’opérateur suivant pour détecter les coins dans une image :

$$K(x, y) = \det(C) - \lambda \text{trace}(C) \quad (1.24)$$

où λ est déterminé par l’expérience, en maximisant la réponse sur un modèle de coin idéal : $\lambda \approx 0,04$.

La matrice C est définie par :

$$C = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \quad (1.25)$$

Le terme $\text{trace}(C)$ constitue l’information “contour” de l’opérateur, tandis que $\det(C)$ contient l’information “coin”. Les valeurs de $K(x, y)$ sont positives au voisinage d’un coin, négatives au voisinage d’un contour et faibles dans une région d’intensité constante.

– **Heitger et Rosenthaler**

Heitger et Rosenthaler [HRH⁺92] ont proposé une autre approche inspirée des mécanismes neuro-biologiques du système visuel. Leur approche consiste à extraire des caractéristiques 1D en convoluant l’image avec des filtres directionnels pairs et impairs dans une même direction. Les caractéristiques 2D sont obtenues en calculant les dérivées premières et deuxièmes sur les caractéristiques 1D.

– **Opérateur de Förstner**

Förstner [FG87] propose une approche basée sur la statistique locale d’une image. L’opérateur extrait des points d’intérêt de types coins ou jonctions dans une image d’intensité, sans faire l’extraction de courbes, contours ou segments de droites, mais en utilisant le gradient d’intensité comme information sur les contours. Förstner utilise un opérateur tel que celui défini par Harris [HS88], avec une estimation de la taille des voisinages autour des jonctions potentielles.

Cette approche a ensuite été améliorée [F94] permettant de classifier le type de structure dans la région d’intérêt, comme coin, anneau, spirale ou texture isotrope. Un test statistique et une étude du champ de gradient permettent cette classification. La méthode peut être divisée en trois étapes : la sélection de fenêtres optimales ne contenant chacune qu’un seul point d’intérêt (basée sur la valeur moyenne du gradient dans une fenêtre de dimensions données), puis la classification du type de point singulier par un test statistique, et enfin la localisation avec une précision sous-pixellique du point optimal dans les fenêtres.

– **SUSAN**

S.M. Smith et J.M. Brady [SB97] ont défini un détecteur de coins et de jonctions ne nécessitant pas le calcul des dérivées de l'image et aucune réduction de bruit. Un masque de pixels de forme circulaire balaye l'image. Le centre de ce masque est appelé *nucleus*, tandis que l'aire du masque représentée par les pixels ayant un niveau de gris voisin du *nucleus* est appelée *USAN* (*Univalve Segment Assimilating Nucleus*). Le fait d'avoir ainsi associé chaque point de l'image à une zone locale de même niveau de gris est à la base du principe de SUSAN. Il est alors possible de détecter à partir de la taille, du barycentre et des moments du second ordre de l'*USAN* à la fois les contours et les coins.

– **SIFT**

La méthode SIFT [Low04] (*Scale Invariant Feature Transform*) est une méthode de description locale de l'image invariante aux changements d'échelle, de perspective, d'éclairage, et particulièrement robuste au bruit dans l'image. Cette approche a été évaluée comme l'une des plus performantes dans une étude comparative sur les descripteurs locaux [MS05]. L'algorithme consiste à extraire un grand nombre de points d'intérêt dans l'image après une succession de traitements (pyramide d'images, différences de gaussiennes, recherche d'extrema). Ensuite chaque point devient un *Key-Point* en lui attribuant l'orientation du gradient en ce point. Enfin on définit pour chaque *KeyPoint* un descripteur, qui est un vecteur à 128 dimensions, représentant l'orientation du gradient dans le voisinage du point. Les descripteurs permettent de corrélérer les points SIFT extraits dans différentes images. L'inconvénient majeur est la taille des descripteurs, rendant ces points difficilement exploitables dans de grandes collections d'images, ou pour du traitement d'image en temps réel.

1.2.2 Mise en correspondance

La mise en correspondance d'informations contenues dans au moins deux images est indispensable pour retrouver l'information 3D manquante, ou pour faire de l'asservissement visuel. L'appariement peut se faire soit entre les primitives extraites dans les images, soit entre tous les points des images, c'est ce que l'on appelle alors l'appariement dense. L'appariement entre les primitives va surtout être utilisé pour trouver la transformation entre deux images, pour calculer une commande, ou pour faciliter un appariement dense qui servira principalement au rendu 3D d'un objet ou d'une scène.

1.2.2.1 Appariement des primitives

L'appariement de points d'intérêt a fait l'objet de nombreux travaux. Il consiste à déterminer quels sont les points d'intérêt qui se correspondent dans les images, c'est-à-dire quelles sont les primitives 2D qui représentent les projections d'une même primitive 3D.

– **Méthodes de masque (*Template*)**

Ces méthodes reposent sur une caractérisation exploitant directement l'information photométrique de l'image. Cette classe de méthodes porte le nom de méthodes

iconiques et permet de caractériser les points de l'image de manière locale. Elles sont directement basées sur la représentation iconique/pixellique. Elles sont souvent connues sous le nom de *template matching* ou de mise en correspondance par la corrélation. On choisit d'abord un template ou une zone d'intérêt $T_1(m, n)$, pour $m \in [0, M - 1]$ et $n \in [0, N - 1]$, dans une image $I_1(i, j)$, pour $i \in [0, I - 1]$ et $j \in [0, J - 1]$, avec $M \ll I$ et $N \ll J$. Ainsi le *template* T_1 est en fait une sous-image de l'image I_1 . Pour trouver la correspondance de T_1 dans une image I_2 de même dimension que I_1 , nous devons calculer une mesure de similarité entre T_1 et une sous-image de I_2 , ayant la même forme et la même dimension que T_1 . Une mesure standard est la distance Euclidienne au carré, donnée par :

$$SSD(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_2(i + m, j + n) - T_1(m, n))^2 \quad (1.26)$$

Si les pixels de I_2 à la position (i, j) ont une correspondance exacte avec ceux de T_1 , alors la distance est nulle. Sinon, la position ayant le maximum de vraisemblance est celle correspondant au minimum de la fonction SSD .

L'avantage de cette approche est que l'effort sur le prétraitement d'image est minimal. Mais les mesures sont très sensibles au bruit, au changement d'intensité, et au déplacement dont la direction de translation n'est pas parallèle au plan de l'image. Ces méthodes sont inapplicables dans le cas de mouvements complexes entre les deux prises de vues (notamment des rotations).

– Méthodes géométriques

D'autres méthodes quantifient l'information par des grandeurs géométriques, comme les contours, les segments ou encore les régions. Elles sont toutefois issues de primitives symboliques, qui entraînent inévitablement une perte d'information, et surtout rendent la qualité de la mise en correspondance fortement dépendante de la structure de l'image. Les méthodes géométriques sont peut-être les plus naturelles car elles procèdent de façon analogue à l'esprit humain. Elles requièrent d'extraire des primitives qui soient significatives, ce qui peut s'avérer difficile si les images sont fortement bruitées. Il s'agit d'identifier dans les images des caractéristiques géométriques communes qui peuvent être, par exemple, des points extrémaux [Thi96], ou des lignes de crête [STA98], mais également des surfaces, des volumes, des repères orientés, ou encore des descriptions structurées mettant en jeu des relations entre éléments. Il est très important de choisir des primitives qui puissent être présentes dans les deux images. Ainsi l'utilisateur peut appairer les primitives manuellement, ou de manière automatique en utilisant des critères de ressemblance.

Une fois le signal visuel réduit à une information géométrique pertinente, d'autres méthodes permettent d'appairer au mieux les différentes primitives en même temps que l'on cherche la transformation spatiale. Les algorithmes de *prédiction-vérification* choisissent un nombre minimum d'appariements probables entre les primitives, afin d'estimer la transformation (comme l'homographie ou la matrice fondamentale) entre ces primitives, et vérifient ensuite la validité de celle-ci sur l'ensemble des primitives. En itérant sur différentes prédictions, il est alors possible d'estimer la

meilleure transformation expliquant les appariements [FB81].

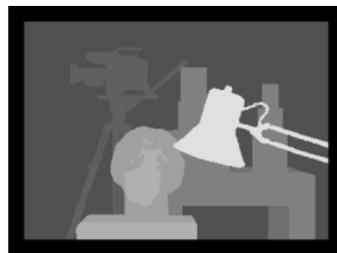
Les attributs photogrammétriques, comme le gradient d'intensité, peuvent être utilisés pour renforcer les contraintes pendant la mise en correspondance. Avec cette approche, une image peut être décrite comme un graphe avec les primitives comme noeuds. Le problème de la mise en correspondance de deux images devient celui de la correspondance entre deux graphes.

1.2.2.2 Méthodes d'appariement dense

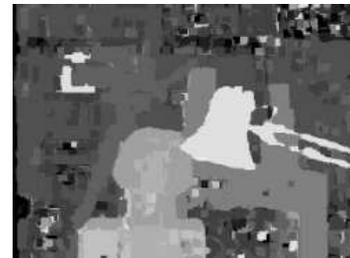
Un grand nombre de méthodes d'appariement dense existe, et elles peuvent être classifiées en deux catégories : les méthodes locales et les méthodes globales. D'autres méthodes appelées algorithmes coopératifs utilisent l'approche locale et globale en même temps. La difficulté est de choisir un algorithme permettant d'obtenir de bons résultats sur des régions sans texture afin d'éviter des zones vides dans la reconstruction, et par rapport aux zones d'occlusion, tout en préservant les détails et les contours. Une étude comparative des performances des différents algorithmes existants [SSZ01] nous a aidés dans le choix de la méthode à adopter pour nos recherches (Fig. 1.5). Ainsi, une méthode d'optimisation globale nommée *Graph-cut* se démarque des autres méthodes sur la préservation des contours et sur l'homogénéité des zones sans texture. Les principaux inconvénients de cette méthode sont son temps de calcul et la précision du réglage de ces paramètres qui dépendent de l'image.



(a) Vue gauche



(b) Carte de disparité théorique



(c) SAD (*Sum of Absolute Differences*)



(d) Scanline Optimization



(e) Dynamic Programming



(f) Graph Cut algorithm

FIG. 1.5 – Comparaisons d'algorithmes d'appariement dense sur les images Tsukuba [SSZ01]

Les méthodes locales

- **NCC** : *Normalized Cross-Correlation*, le critère le plus utilisé, mais aussi le plus long à calculer. C'est la somme des produits entre les termes correspondants des fenêtres de corrélation, normalisée par le produit des moyennes quadratiques calculées pour chacune des fenêtres.

Valeur du critère au pixel (u, v) pour une disparité d :

$$NCC(u, v, d) = \frac{\sum_{i,j} (I_1(u+i, v+j) - \bar{I}_1) \cdot (I_2(u+d+i, v+j) - \bar{I}_2)}{\sqrt{\sum_{i,j} (I_1(u+i, v+j) - \bar{I}_1)^2 \cdot (I_2(u+d+i, v+j) - \bar{I}_2)^2}}$$

La fenêtre de corrélation est centrée autour du pixel (u, v) , I_1 et I_2 représentent les images mises en correspondance, d est la disparité, et \bar{I}_1 et \bar{I}_2 correspondent aux moyennes de l'intensité des pixels sur la fenêtre de corrélation.

- **ZNCC** : *Zero mean Normalized Cross Correlation*, la méthode ressemble à NCC mais on fait une corrélation croisée entre les écarts à la moyenne. C'est aussi un critère invariant aux variations uniformes de luminance dans une des images.
- **SSD** : *Sum of Squared Differences*, ce qui correspond à la somme des différences quadratiques entre termes correspondants des deux fenêtres.

$$SSD(u, v, d) = \sum_{i,j} (I_1(u+i, v+j) - I_2(u+d+i, v+j))^2$$

- **ZSSD** : *Zero mean Sum of Squared Differences*, la méthode est identique à SSD sauf que, au préalable, pour chaque fenêtre de corrélation, la moyenne des termes est d'abord calculée et soustraite à chaque terme : la différence est donc faite entre les écarts à la moyenne des termes, et non entre les termes eux-mêmes.
- **SAD** : *Sum of Absolute Differences*, c'est la somme de la valeur absolue des différences entre termes correspondants des deux fenêtres.

$$SAD(u, v, d) = \sum_{i,j} |I_1(u+i, v+j) - I_2(u+d+i, v+j)|$$

- **ZSAD** : *Zero mean Sum of Absolute Differences*, c'est une méthode semblable à SAD, mais on fait la différence entre écarts à la moyenne. C'est donc un critère invariant aux variations uniformes de luminance dans une des images.
- **Mise en correspondance par des invariants locaux** : des méthodes d'extraction de primitives caractérisent les points extraits dans l'image par un vecteur d'attributs invariants à un certain nombre de transformations comme la rotation, le changement d'éclairage ou d'échelle. La méthode SIFT [Low04] est l'une de ces méthodes, particulièrement robuste au bruit et à un certain nombre de changements dans l'images. Le calcul de ces points est généralement très coûteux en terme de temps suivant la

taille des vecteurs (128 dimensions pour les descripteurs de la méthode SIFT). Mais le nombre de points extraits et mis en correspondance est au final très restreint. Tous les points ainsi détectés sont ensuite joints par une triangulation de type Delaunay. L'espace de recherche de nouveaux appariements est donc restreint par la surface des triangles dont les sommets ont été mis en correspondance. Les nouveaux points caractéristiques extraits et appariés dans les triangles sont à nouveau joints par triangulation, et ainsi de suite jusqu'à obtenir une carte d'appariement dense. Cette méthode est basée sur les techniques récursives de prédiction et de vérification d'hypothèses, où les appariements valides sont utilisés pour générer des hypothèses d'appariement (propagation basée sur les relations de voisinage entre primitives).

Les méthodes globales

Les méthodes globales utilisent des contraintes non-locales afin d'augmenter la robustesse dans les régions locales de l'image, où l'appariement échoue dans des zones de texture uniforme, ou à cause des occlusions et des défauts d'illumination. Par contre l'inconvénient majeur de ces méthodes est la plus grande complexité des calculs. Ces méthodes sont représentées sous forme de problème d'optimisation, cherchant à assigner les disparités qui minimisent une fonction de coût globale. Appartiennent à ce groupe les méthodes de :

- **Programmation dynamique**

La programmation dynamique est une méthode assez générale pour résoudre des problèmes d'optimisation non-linéaire avec contraintes, quand les variables ne sont pas toutes interdépendantes. Un tel problème peut être décomposé en une séquence de sous-problèmes d'optimisation avec une seule variable, ou de façon plus générale avec un sous-ensemble de variables. La programmation dynamique est équivalente à la recherche d'un chemin optimal dans un graphe, par la minimisation d'une fonction de coût. L'application de cette technique à la mise en correspondance de primitives dans une paire d'images stéréo a été faite par Ohta et Kanade [OK85].

- **Flot de graphe** (*Graph-cut*)

Roy and I. J. Cox [RC98] ont été les premiers à généraliser l'algorithme *Graph-cut* pour construire des cartes de disparités. Le but de cette méthode est de trouver la coupure d'un graphe qui possède un coût minimum parmi toutes les coupures possibles, en minimisant une fonction d'énergie. Les terminaux du graphe représentent les disparités pouvant être associées à un pixel de l'image, et les différents noeuds correspondent aux pixels de l'image. Pour définir la fonction d'énergie, on introduit une fonction de coût basée sur un critère de photoconsistance (mesure sur la similarité de l'intensité des pixels) appelé le terme des données (*data term*). Un second terme, appelé terme d'homogénéité ou d'uniformité spatiale (*smoothness term*), pénalise les discontinuités entre pixels voisins. Cette méthode permet donc d'obtenir une carte de disparités avec des zones homogènes tout en préservant les contours. Le défaut majeur de cette méthode est d'aplatir les objets.

- **Optimisation ligne par ligne** (*Scanline Optimization*)

Cette methode globale d’optimisation 1D n’utilise que les termes horizontaux de lissage (*smoothness terms*), en résolvant le même problème d’optimisation que l’algorithme de graph-cut (sauf que les termes verticaux de lissage sont donc ignorés)[SSZ01].

- **Relaxation**

C’est un algorithme itératif et localement parallèle, qui améliore une solution en minimisant localement l’énergie [RHZ76, KR79, HZ83]. L’implémentation de ce genre de méthode est simple, mais la vitesse de convergence est lente.

1.2.2.3 Contraintes géométriques

En dehors de la contrainte photométrique permettant la mise en correspondance de deux pixels à partir de leur niveau de gris, en s’appuyant sur l’hypothèse que les projections d’un même point de l’espace tridimensionnel ont des intensités lumineuses comparables, d’autres contraintes d’ordre géométrique améliorent l’appariement des points :

- **Géométrie épipolaire**

Comme nous l’avons vu dans la partie précédente, la contrainte épipolaire permet de réduire le domaine de recherche d’un point dans l’autre image, à une ligne appelée ligne épipolaire. Ceci n’est vrai que dans le cas idéal, mais l’échantillonnage et l’estimation de la matrice fondamentale modifient la zone de recherche d’une ligne à un bandeau. Si la géométrie épipolaire permet de réduire l’espace de recherche de mise en correspondance, elle permet aussi d’éliminer les mauvais appariements. Ainsi l’estimation de la matrice fondamentale et par la même occasion l’élimination des mauvais appariements peut se faire en utilisant l’algorithme de RANSAC (*RANdom Sample Consensus*). Cet algorithme proposé par Fischler et Bolles [FB81] choisit d’une manière aléatoire des échantillons d’appariements de primitives du modèle, qui sont ensuite utilisés pour calculer les paramètres inconnus d’une transformation, qui est dans notre cas la matrice fondamentale. Cette transformation est immédiatement évaluée sur le reste des primitives. Si elle donne un consensus satisfaisant, l’algorithme se termine. Dans le cas contraire, l’échantillonnage continue jusqu’à trouver une transformation satisfaisante.

- **Ordonnement**

Pour les surfaces opaques l’ordre des pixels voisins est préservé sur la ligne épipolaire correspondante [Fal94, KPG00]. Ainsi, si \mathbf{p}_1 et \mathbf{p}_1^* sont les projections d’un même point χ_1 , alors tous les points à droite de \mathbf{p}_1 sont à droite du point \mathbf{p}_1^* (Fig. 1.6).

- **Unicité**

La correspondance doit être bidirectionnelle tant qu’il n’y a pas d’occultation sur l’une des images. Cette contrainte se justifie dans le cas de surfaces opaques, sauf en cas d’alignement de deux points avec le centre optique de la caméra. Ainsi, un point 3D dont la projection est un pixel sur la première image a une projection qui

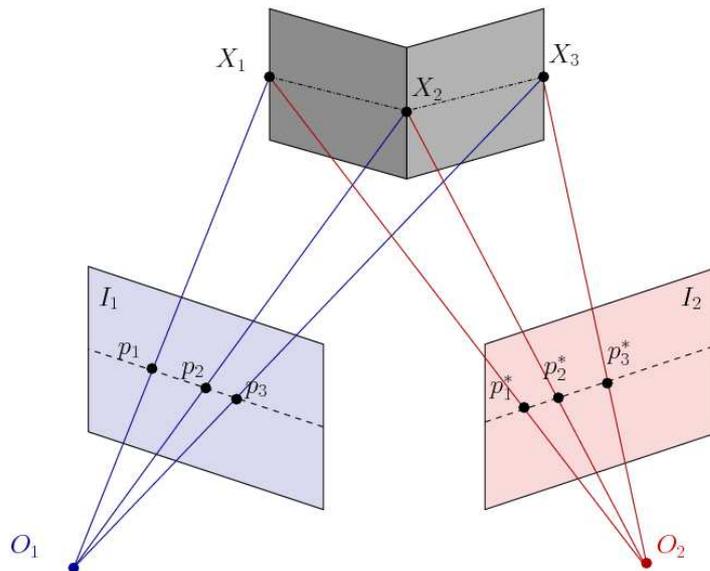


FIG. 1.6 – Contrainte d'ordonnancement

est aussi un pixel sur la seconde image. Cette contrainte permet donc de repérer les mauvais appariements et les occultations.

– Base stéréo

La base stéréo est l'espace entre deux vues stéréoscopiques. Une base est étroite lorsque sa valeur est faible par rapport à la distance entre l'objet et les différents points de vue, comme par exemple entre les images d'une séquence vidéo d'une caméra en mouvement. Les avantages sont la facilité de mise en correspondance des primitives extraites dans les images, et les faibles régions d'occlusion. Mais l'inconvénient est le faible angle de triangulation pour déterminer les profondeurs. Inversement, une base stéréo large, par exemple en prenant quelques images avec des points de vue très différents, permettra une meilleure précision sur la profondeur des points, mais la mise en correspondance sera plus difficile, et davantage de zones d'occlusions vont apparaître. Une méthode permet de combiner les avantages des deux approches, en fusionnant les différentes cartes de disparités calculées entre les points de vues les plus proches pour déterminer un modèle 3D (*multi viewpoint linking*) [KPG98]. Ainsi on obtient une carte de profondeur très dense, avec une meilleure précision sur les profondeurs, et moins de zones d'occlusions.

1.2.3 Suivi visuel

L'extraction et l'appariement de primitives dans les images ne sont que l'étape d'initialisation du processus permettant d'asservir visuellement une caméra entre une position courante et une position de référence dans l'espace cartésien. Il faut donc ensuite pouvoir suivre visuellement les primitives extraites dans l'image courante, jusqu'à ce que la fonction de tâche ait convergé vers zéro.

Les méthodes de suivi visuel peuvent dépendre des connaissances a priori que l'on a sur l'objet, comme par exemple le modèle 3D ou la texture. De plus, le suivi visuel pouvant être considéré comme un problème d'optimisation, les méthodes diffèrent suivant la fonction et la manière d'optimiser. Enfin ces méthodes varient également selon que l'on désire suivre des primitives ou des régions.

1.2.3.1 Méthodes de corrélation

Les méthodes de suivi par corrélation considèrent un certain voisinage \mathcal{W} d'un point \mathbf{p} dans une image \mathcal{I}_1 , et recherchent dans une seconde image \mathcal{I}_2 la position du point \mathbf{p}' , dont le voisinage \mathcal{W} permet de maximiser une fonction de similarité (mesure de corrélation), ou minimiser une fonction de dissimilarité (emploi de distances). Ces fonctions ont été décrites dans le paragraphe sur les méthodes locales d'appariement dense (cf. §1.2.2.2 page 21). Ce type de mesures corrélatives est également largement utilisé en stéréovision [BC01]. Le choix de la taille de la zone de recherche s'avère crucial au bon fonctionnement de la méthode de suivi. Ainsi, elle doit être suffisamment large pour ne pas perdre les points lorsque les déplacements inter-images sont importants, mais elle doit être également suffisamment restreinte de manière à éviter les ambiguïtés éventuelles (par exemple en cas de répétitions d'un même motif). Pour finir, elle ne doit pas être trop coûteuse en temps de calcul.

L'inconvénient de ces méthodes de corrélation réside dans leur faible prise en compte des déformations géométriques dues aux mouvements 3D de la caméra ou de l'objet. Généralement, de simples translations ou similitudes sont considérées. Pour y remédier, les changements d'échelle peuvent être compensés par modélisation [SMR96], ou par une variation de la taille de fenêtre, telle que dans [KO94, CS00], permettant en outre de prendre en compte les variations de forme de l'objet à suivre. Afin de contourner le problème de déformation, [MIB04] propose une méthode simple de remise à jour de la primitive à suivre, tout en évitant les dérives temporelles. Ainsi, la primitive n'est mise à jour que si les erreurs de corrélation sont suffisamment faibles par rapport à la primitive initiale.

1.2.3.2 KLT

Cette approche a été introduite par les travaux précurseurs dus à Lucas et Kanade [LK81]. Les auteurs proposent d'estimer le déplacement en translation de certaines régions d'intérêt dans l'image en utilisant la minimisation de Gauss-Newton. Ils supposent avoir une estimation préalable du déplacement et déterminent de manière itérative les incréments nécessaires pour minimiser une fonction de coût. Contrairement aux méthodes de suivi par corrélation, les méthodes différentielles telles que le KLT (Kanade, Lucas, Tomasi) ne requièrent pas de recherche exhaustive. Cette méthode est basée sur l'approximation au premier ordre de la variation du signal image par rapport aux incréments du déplacement. Elle est caractérisée par le fait que la direction de descente est estimée par un Jacobien calculé à l'état courant de la minimisation, et par le fait que l'incrément du déplacement se fait d'une manière additive. D'où son appellation "forward additional approach" par [BM01]. Cette approche a été généralisée aux transformations affines par [ST94], qui par la même occasion propose une manière de sélectionner les "bon-

nes” régions d’intérêt à suivre. Récemment dans [Bou99], une approche pyramidale a été proposée pour l’implémentation de cet algorithme. Elle permet d’accélérer l’algorithme et d’avoir une zone de convergence plus grande. C’est cette même méthode qui est actuellement implémentée dans la très populaire librairie de traitement d’images et de vision par ordinateur “OpenCV”. Encore aujourd’hui, cette technique KLT, que nous utiliserons dans la suite de ce manuscrit, est réputée très performante [TS04], et il en existe de nombreuses variantes dont une comparaison est menée dans [BM04a].

1.2.3.3 ESM

La méthode ESM (Efficient Second-order Minimization) proposée par Benhimane et Malis [BM04b], est une approche de suivi visuel d’objets plans grâce à leurs textures dans l’image en utilisant une minimisation efficace au second ordre. Cette méthode résout un problème d’optimisation en supposant que les paramètres de transformation varient localement, d’une manière quadratique par rapport à l’erreur des intensités lumineuses mesurées. Les paramètres de transformation estimés sont les paramètres de transformation projective dans l’image. Ce suivi visuel a des propriétés de convergence supérieures à celles des méthodes de suivi visuel du premier ordre de type Gauss-Newton (domaine, taux et fréquence de convergence) [BM04b], telles que Forward Additional [LK81], Forward Compositional [SS00], Inverse Additional [HT97] et Inverse Compositional [BM01]. Une minimisation utilisant des approximations au second ordre (Newton) a une convergence rapide, localement quadratique, mais le temps de calcul est plus important qu’une méthode du premier ordre, car on calcule toujours un Jacobien mais avec en plus des matrices Hessiennes. D’un autre côté, une minimisation utilisant des approximations au premier ordre (Gradient, Gauss-Newton, Levenberg-Marquandt) possède une convergence plus lente, au mieux super-linéaire. L’approche ESM possède l’efficacité d’une méthode du second ordre, alliée à la rapidité de convergence d’une méthode du premier ordre. Le principal inconvénient de cette méthode pour notre application de suivi visuel est la limitation du domaine d’application au suivi de cibles planaires, ce qui est difficilement envisageable dans une scène naturelle. Cependant en raison de l’efficacité et de la robustesse de la méthode, nous l’avons tout de même envisagée pour des applications en laboratoire sur des cibles planaires.

1.3 La reconstruction 3D

La reconstruction 3D de surfaces à partir d'une séquence d'images est un sujet de recherche très actif dans la communauté de la vision par ordinateur. Il existe de nombreuses méthodes de reconstruction 3D, mais toutes ne sont pas directement applicables sur nos images sous-marines. Les approches existantes diffèrent beaucoup selon les techniques utilisées ainsi que par les connaissances *a priori* du système et de l'environnement. On peut citer les méthodes pour modéliser la végétation comme les arbres par exemple [SRDT01], les approches de numérisation 3D de visage par stéréoscopie active [BAC06], ou la modélisation 3D de bâtiments pour la création de bases de données urbaines 3D [PPPD05, PBT07]. Les hypothèses portent sur la rigidité et l'opacité d'un objet statique. Les paramètres intrinsèques sont connus après la phase de calibration et les paramètres extrinsèques sont estimés grâce à notre méthode d'acquisition, comme nous le verrons dans le deuxième chapitre. La modélisation d'un paysage ou d'une scène complète sous-marine est souvent rendue trop difficile par le manque de visibilité, nous nous limitons donc à la reconstruction d'objets de faible volume (de l'ordre du mètre cube).

On peut répertorier une grande partie des méthodes de reconstruction dans ces cinq catégories (mais il s'agit évidemment d'une liste non exhaustive) :

- Reconstruction à partir de la géométrie épipolaire ;
- Reconstruction à partir de la texture ;
- Reconstruction à partir des silhouettes ;
- Reconstruction à partir du mouvement de la caméra ;

On n'abordera pas ici les méthodes de reconstruction à partir de jeux de lumière (ombres et lumière structurée), ni celles basées modèles, sachant que l'on ne dispose pas d'information *a priori* sur le modèle de la scène sous-marine à reconstruire.

1.3.1 Reconstruction à partir de la géométrie épipolaire

Pour illustrer les méthodes basées sur la géométrie épipolaire, nous présentons une méthode générale de reconstruction par extraction de points à partir de deux vues. Cette méthode proposée par Hartley et Zisserman [HZ00], se décompose en plusieurs phases :

- **Extraction des points d'intérêts :**
Les points d'intérêts sont des primitives extraites dans l'image par un détecteur de points (cf. §1.2.1 page 15).
- **Mise en correspondance :**
La mise en correspondance consiste à localiser dans les images, les projections de la même entité de la scène, c'est-à-dire à apparier les points d'intérêts (cf. §1.2.2 page 18).
- **Reconstruction projective :**
Dans une image perspective, seules des propriétés projectives telles que le birapport, l'incidence et la colinéarité sont préservés. Il n'est donc pas possible de reconstituer toutes les propriétés de la scène telles que les distances ou les angles, sans informa-

tions supplémentaires autres que les images.

Si on ne connaît pas les paramètres de la caméra, on peut tout de même retrouver la position 3D des points à travers une transformation projective. Une telle projection ne conserve ni le parallélisme, ni la métrique de la scène.

Considérons la matrice fondamentale \mathbf{F}_{12} et l'épipole \mathbf{e}_{21} d'une paire d'images I_1 et I_2 . L'épipole \mathbf{e}_{21} représente la projection du centre optique O_1 sur l'image I_2 . Les matrices de projection \mathbf{P}_1 et \mathbf{P}_2 d'un point de l'espace sur les images I_1 et I_2 sont données par :

$$\begin{cases} \mathbf{P}_1 = (\mathbf{I} & \mathbf{0}) \\ \mathbf{P}_2 = ([\mathbf{e}_{21}]_{\times} \mathbf{F}_{12} & \mathbf{e}_{21}) \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\alpha}^{\top} & \alpha_4 \end{pmatrix} \mathbf{G} \end{cases} \quad (1.27)$$

où $[\mathbf{e}_{21}]_{\times}$ est la matrice antisymétrique du vecteur \mathbf{e}_{21} . Le vecteur $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}^3$ et $\alpha_4 \neq 0 \in \mathbb{R}$ sont choisis de manière à ce que les matrices de projection soient numériquement bien conditionnées. Enfin, la matrice $\mathbf{G}(4 \times 4)$ est une matrice arbitraire non-singulière représentant une déformation projective, pouvant être égale à la matrice identité.

Une méthode de reconstruction projective par factorisation, à partir de plusieurs images orthographiques, a été introduite par Tomasi et Kanade [TK92], puis Sturm et Triggs l'ont étendue aux caméras perspectives [Stu97b]. L'idée est de factoriser une matrice contenant les coordonnées de tous les points image. Cette matrice doit être idéalement de rang 3 afin de pouvoir être factorisée en deux matrices, l'une contenant les coefficients des matrices de projection, l'autre contenant les coordonnées des points 3D reconstruits de manière affine.

– **Reconstruction euclidienne :**

Pour reconstruire la structure euclidienne de la scène, il faut connaître les paramètres de calibration de la (ou les) caméra(s). La calibration permet d'établir une correspondance entre les dimensions réelles et les dimensions image. Elle peut se faire à partir de connaissances géométriques dans la scène, telles que des droites parallèles, ou en exploitant les propriétés de certains déplacements de la caméra (rotation pure par exemple), ou encore le plus souvent grâce à une mire de calibration. Le groupe des transformations euclidiennes est un sous-groupe des transformations affines, lui-même un sous-groupe des transformations projectives (Fig. 1.7). La transformation euclidienne respecte les angles et les distances en plus du parallélisme présent dans la transformation projective.

– **Visualisation :**

Afin d'améliorer la résolution de la surface de la reconstruction 3D, et donc le rendu visuel, il est nécessaire de calculer une reconstruction dense. Cette étape commence par une rectification des images, correspondant à une transformation permettant l'application d'un algorithme d'appariement dense entre les pixels. Pour simplifier et homogénéiser la structure de points obtenue, on détermine ensuite un maillage par une triangulation de Delaunay (Fig. 1.8). Cette triangulation est obtenue en reliant par une arête les points dont les régions correspondantes dans le diagramme

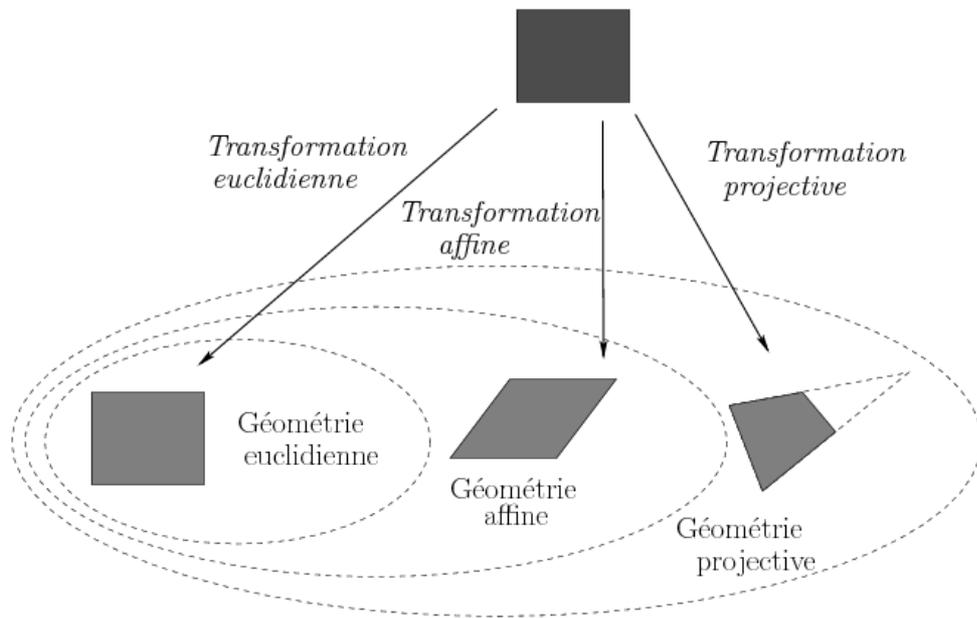


FIG. 1.7 – Géométries affine, projective et euclidienne

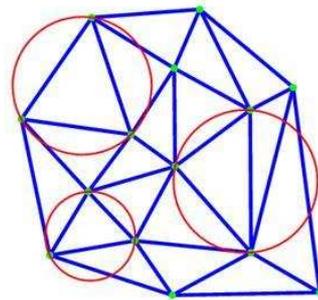


FIG. 1.8 – Triangulation 2D de Delaunay

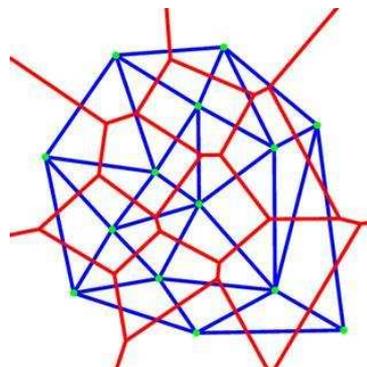


FIG. 1.9 – Triangulation de Delaunay (bleu) et diagramme de Voronoï (Rouge). C'est le dual de la triangulation de Delaunay. Le polygone centré sur un point p de E est à la frontière de l'ensemble P des points du plan, plus proches de p que tout autre point de E .

de Voronoï sont adjacentes (Fig. 1.9). Si l'on considère un ensemble E de n points du plan, la triangulation de Delaunay T de E , est la triangulation telle qu'aucun triangle de T ne contient un autre point de E à l'intérieur de son cercle circonscrit. Enfin on plaque sur le modèle 3D la texture correspondante à une des images de la séquence.

Les méthodes basées sur la géométrie épipolaire sont les plus adaptées à notre cas comme nous le verrons par la suite, étant donné que l'on utilise déjà la géométrie épipolaire pour initialiser l'asservissement visuel et que l'on peut estimer les positions des caméras. D'autres méthodes utilisent le tenseur trifocal qui peut être considéré comme une extension de la géométrie épipolaire à trois images par rapport à la matrice fondamentale.

1.3.2 Reconstruction à partir de la texture

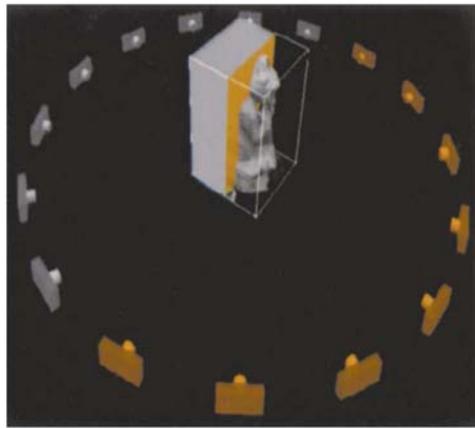
Les méthodes de reconstruction à partir de la texture sont des méthodes voxéliques de "sculptage" d'un volume (ou *space carving*) dans l'espace suivant un critère de photoconsistance [KS00] (Fig. 1.10). Les avantages de ce genre de méthode sont la simplicité des calculs et le texturage immédiat en plus du modèle 3D. Mais il y a également plusieurs inconvénients comme la sensibilité aux changements d'éclairage et le temps de calcul élevé pour obtenir une bonne définition. Afin d'améliorer le rendu de la surface constituée de petits voxels (souvent aspect rugueux), on lisse par une méthode de *marching cubes*, qui crée pour chaque cube une maille triangulaire se rapprochant de l'iso-surface, puis on calcule les normales à la surface à chaque sommet du triangle, permettant ainsi de faire l'ombrage de l'objet par la suite.

D'autres méthodes voxéliques se basent sur *l'évolution de surface par EDP*, balayant l'espace avec une frontière mouvante représentée par une équation aux dérivées partielles (EDP), contrairement à la méthode de sculptage qui balaye l'espace voxélique plan par plan dans des directions alternantes jusqu'à la convergence. L'évolution de cette frontière est pilotée par la mesure de consistance de couleur comme précédemment. Cette approche est mathématiquement beaucoup plus sophistiquée, mais pas beaucoup plus complexe à mettre en œuvre.

Ces méthodes basées sur le calcul d'un critère de photoconsistance nécessitent d'avoir une lumière constante et homogène tout autour de l'objet, ce qui n'est absolument pas notre cas dans les fonds sous-marins.

1.3.3 Reconstruction à partir des silhouettes (aspect)

Cette approche utilise les contours occultants, correspondant à la silhouette de l'objet dans l'image. L'idée est de délimiter un volume dans l'espace se rapprochant le plus possible de l'objet observé (Fig. 1.11). Cette approximation est appelée "enveloppe visuelle" (*visual hull*). Il existe deux manières de déterminer cette enveloppe visuelle : soit par intersection des "cônes d'observation" (volume dans l'espace délimité par les contours dans l'image), soit en "sculptant" un volume de départ, en sélectionnant les voxels suivant leur



(a) La photoconsistance des voxels sur le plan en orange, est testée dans les caméras actives également en orange, où les voxels sont visibles



(b) Reconstruction d'une main par *space carving*. Le modèle comporte 112 000 voxels

FIG. 1.10 – Méthode de reconstruction voxélique par *space carving* (illustrations extraites de [KS00])

reprojection dans les images, à l'intérieur ou à l'extérieur du contour [BB97, Lau95].

Ces techniques sont intéressantes sur des structures de petites ou moyennes dimensions, pour lesquelles il est possible d'utiliser une table tournante ou de déplacer une caméra autour de l'objet, et de définir un fond pouvant être facilement soustrait dans les images afin de mieux définir les contours de l'objet.

Baser la reconstruction 3D sur les contours de l'objet semble donc être difficilement envisageable pour une scène naturelle sous-marine. La méthode semble simple et robuste aux changements d'intensité, mais on ne peut au mieux que retrouver l'enveloppe convexe de l'objet, mais pas les concavités.

1.3.4 Reconstruction à partir du mouvement

Hormis certaines configurations dégénérées, le mouvement de la caméra dans l'espace tridimensionnel se projette en un mouvement 2D dans la séquence d'images capturées. Ce mouvement 2D peut être représenté dans le plan image comme un champ vectoriel de déplacement. En mesurant ce champ de vecteurs on peut estimer la carte de disparité, et retrouver la structure de l'objet observé (cf. Fig. 1.12). Les champs de vecteurs peuvent être approximés comme des champs de vitesse (flot optique).

Ainsi, la reconstruction de la structure à partir du mouvement (*Shape from motion*) consiste à prendre à l'aide d'une caméra, une séquence d'images de l'objet placé par exemple sur une table tournante, ce qui correspond à un environnement contrôlé. Ensuite pour chaque image, on estime le mouvement apparent de certains points (ou de tous les points de l'image), puis on retrouve la forme 3D à l'aide des vitesses estimées. Le mouvement est supposé très faible entre chaque image, et la scène est statique afin que le

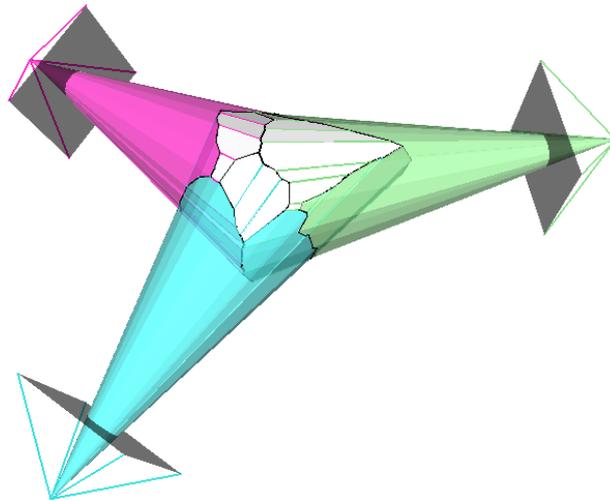


FIG. 1.11 – Reconstruction de l'enveloppe visuelle d'un objet par intersection des cônes d'observation dans l'espace

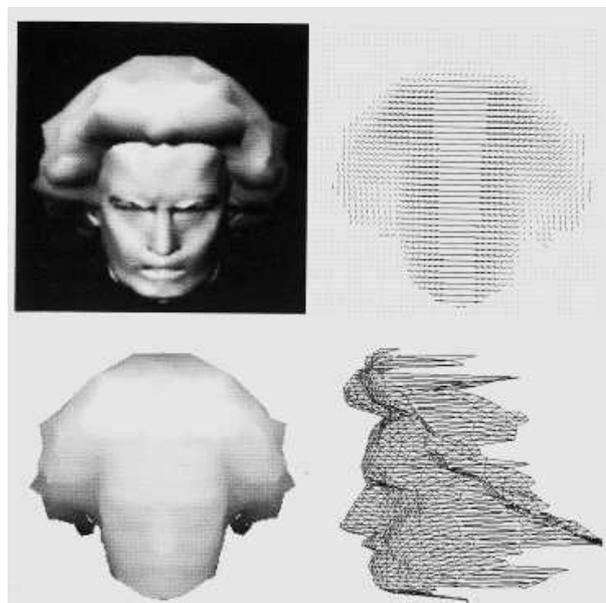


FIG. 1.12 – Reconstruction de la structure d'un objet (en haut à gauche) à partir du champ vectoriel (en haut à droite), permettant de définir une carte de profondeur (en bas à gauche) pouvant ensuite être représentée par un maillage 3D (en bas à droite) [KSK⁺98]

déplacement des points dans l'image ne soit dû qu'au mouvement de la caméra. L'avantage de ce genre de méthode est de pouvoir facilement mettre en correspondance les pixels entre les images par des méthodes de suivi visuel, pouvant exploiter l'information du mouvement effectué pour prédire les disparités dans l'image suivante. Mais l'inconvénient est une plus grande sensibilité au bruit par rapport à une méthode stéréo, car l'espacement entre deux images consécutives étant très faible, le calcul de la disparité est instable.

1.4 L'asservissement visuel

Pour connaître les positions des prises de vue servant à la reconstruction, nous avons besoin de contrôler le mouvement des caméras, comme nous l'avons vu précédemment dans l'introduction. L'approche choisie est de faire ce contrôle par asservissement visuel puisque nous disposons d'un robot équipé d'un bras manipulateur et de caméras stéréo. Les méthodes d'asservissement visuel ont pour objectif de contrôler le mouvement d'un dispositif physique ou virtuel, à partir des informations acquises par un capteur visuel embarqué ou non sur ce dispositif, et observant la scène. Elles s'appuient sur des schémas de contrôle en boucle fermée sur les informations visuelles. La commande résultant des informations visuelles est envoyée au dispositif. L'approche imaginée est d'utiliser la vision active pour la phase d'acquisition des images, l'asservissement visuel des caméras assurant le contrôle du déplacement du manipulateur portant la tête stéréo le long d'une trajectoire prédéfinie.

1.4.1 La vision active

La vision active n'utilise pas de capteurs actifs (lasers par exemple), mais utilise des capteurs passifs de manière active. Ces techniques tirent leur origine d'une tentative de simulation du système visuel humain. Concernant la vision humaine, le mouvement des yeux et de la tête, l'adaptation des pupilles aux variations d'illumination, jouent un rôle important dans la perception visuelle. C'est cette faculté d'adaptation que la vision active tente de recréer. En fonction de la tâche à effectuer et/ou de stimuli externes, un système de vision active peut être amené à modifier les paramètres extrinsèques et/ou intrinsèques d'une ou plusieurs caméras (position, orientation, zoom, etc.), mais aussi, à se focaliser sur certaines régions d'une image pour effectuer un traitement particulier. La vision active prend donc à la fois le contrôle du matériel et des ressources logicielles allouées au système [Swain 93]. Cette répartition et ce contrôle intelligent de l'ensemble des ressources disponibles ont pour objectif une optimisation du processus de perception, que ce soit au niveau de la qualité des résultats obtenus, de la complexité des algorithmes mis en jeu, ou de la quantité d'information extraite de la séquence d'images.

1.4.2 La vision stéréoscopique

Le monde réel perçu par une machine de vision artificielle est un univers en deux dimensions. En effet, une caméra effectue une opération de projection perspective \mathcal{P} qui transforme le monde tridimensionnel de la scène en une représentation bidimensionnelle, causant la perte de l'information de profondeur. Si cette transformation de projection est connue, il est possible, étant donné un point physique de la scène, de connaître précisément la position de sa projection dans l'image. En revanche, étant donné un point de l'image, il existe une infinité de points de la scène portés par une droite qui vérifie la transformation inverse [LL86]. Un complément d'information est nécessaire pour déterminer la coordonnée tridimensionnelle manquante. La solution se trouve dans la combinaison de plusieurs images prises de points de vue différents. La stéréoscopie est définie comme un procédé donnant l'impression de relief à partir d'un couple d'images enregistrées selon des

points de vue différents. En particulier, la vision binoculaire est l'interprétation de deux vues distinctes de la scène afin de résoudre l'ambiguïté de la profondeur. Chez l'homme, par exemple, les deux yeux sont écartés d'une dizaine de centimètres et perçoivent, de ce fait, deux images assez semblables mais différentes de la même scène. C'est dans cet écartement même faible que nous puisons l'information permettant de reconstruire la dimension perdue. Plus un point est proche de nous et plus ses projections sur les deux rétines sont distantes. La mesure de cette distance relative permet à notre cerveau d'estimer par triangulation les coordonnées du point dans l'espace. Associer ainsi deux à deux toutes les projections rétinienne se correspondant, c'est-à-dire les apparier, nous permet d'apprécier le relief. Un ordinateur, qui reçoit des images prises par deux caméras, fera de même. La stéréoscopie est utilisée pour évaluer rapidement le profil des objets en trois dimensions permettant ainsi une planification plus efficace lors de trajectoires à suivre. Dans notre approche d'acquisition d'images à l'aide d'un banc stéréo porté par un robot manipulateur, qui est détaillée dans la prochaine partie, nous utiliserons le mouvement du bras, pour piloter de façon active la géométrie de l'acquisition des couples d'images.

1.4.3 Contrôle d'une caméra par fonction de tâche

1.4.3.1 Formalisme de la fonction de tâche

L'objectif de commande consiste à contrôler le mouvement de l'effecteur E du robot pour l'amener en un point physique C de l'environnement suivant une certaine orientation relativement au référentiel de description R . La posture désirée de l'effecteur dans l'environnement, en termes de position et d'orientation, notée $(x_C)_R$ est représentée par la configuration d'un repère cible à atteindre. Le repère cible est supposé fixe par souci de simplicité, cette hypothèse n'altérant en rien la généralité du problème.

Définissons par $(\Delta x(t))_R = (x_C)_R - (x_E(t))_R$ une mesure de la position et de l'orientation relatives du repère effecteur et du repère cible exprimée dans R . Cette fonction caractérise précisément la tâche étant donné que son annulation garantit le bon positionnement et l'orientation de l'effecteur sur la cible. Elle est appelée fonction de tâche ou fonction de sortie [SBE91]. La fonction de tâche est un outil mathématique bien utile pour établir l'influence mutuelle de plusieurs tâches, quel que soit le capteur ou la stratégie de commande associés. L'intégration de l'asservissement visuel dans l'approche générale de la fonction de tâche permet de résoudre de manière efficace et élégante le problème de tâche hybride, combinant plusieurs objectifs à atteindre, comme par exemple des opérations de suivi de trajectoires pour des applications d'inspection, d'évitement des butées et singularités du robot ou d'évitement d'obstacles [Mar04]. Le problème du contrôle robotique réside alors dans l'élaboration d'une loi de commande permettant l'annulation de cette fonction d'erreur. L'effecteur étant positionné et orienté dans l'environnement par l'intermédiaire du vecteur des consignes articulaires $q_E(t) \in \mathbb{R}^{n_a}$, réaliser cette tâche équivaut donc formellement à déterminer la configuration articulaire q_C du système mécanique associée à la posture désirée de l'effecteur $(x_E)_R = (x_C)_R$.

Différentes configurations d'agencement du dispositif physique et du capteur visuel sont possibles selon que le capteur est embarqué sur le dispositif (l'oeil est dans la main et observe la scène *eye-in-hand*), ou déporté (l'oeil regarde la scène et éventuellement la

main *eye-to-hand*). Les méthodes d'asservissement visuel peuvent être classifiées suivant l'espace d'état dans lequel s'effectue le contrôle du robot [SW80] :

- **Asservissement visuel 3D** : la position de l'effecteur du robot est estimée à partir des informations visuelles, le contrôle se fait alors dans l'espace cartésien [WCG96, EGAG96, MDGD97, TSSN97]. Ce genre de méthode repose très généralement sur la connaissance a priori d'un modèle (3D ou 2D) de l'objet et des paramètres intrinsèques de la caméra. Toutefois, des techniques de reconstruction 3D par vision dynamique permettent d'estimer le modèle de l'objet d'intérêt, ou de localiser la caméra à partir de mesures de mouvement 2D ou 3D, élargissant ainsi les tâches considérées.
- **Asservissement visuel dans l'espace articulaire** : la configuration articulaire du robot est observée par une (ou plusieurs) caméra(s) déportée(s) [SBJ90, CKS94].
- **Asservissement visuel dans l'espace du capteur** : la fonction de tâche est construite directement à partir des informations du capteur [MCSP02].

Dans notre cas, nous nous intéressons à la troisième méthode. Dans l'approche référencée image [Wei84, FM81, ECFP92, HHC96], la fonction de tâche représentée par une erreur à minimiser, est calculée à partir de la différence entre les informations visuelles courantes (images de la caméra asservie) et désirées (image de référence). Cette approche appelée *asservissement visuel 2D* [ECFP92, Cha90, Has93, PKK93], permet de contrôler les mouvements de la caméra en construisant la fonction de tâche de manière à utiliser directement les informations de l'image sans passer par une estimation (ou calcul) explicite des erreurs de position dans l'espace cartésien. Une approche plus récente, mentionnée sous les termes d'*asservissement 2D 1/2* ou *hybride*, consiste à inclure dans le signal d'erreur à minimiser à la fois les variations de position dans l'image des indices visuels mais également une autre composante qui est souvent issue de la profondeur [Mal98, MC00]. Le contrôle est effectué en partie dans l'image et en partie dans l'espace cartésien. L'avantage majeur de cette approche est que la connaissance du modèle géométrique 3D de l'objet n'est plus nécessaire.

1.4.3.2 La matrice d'interaction

Dans le cas d'objets rigides, la relation vision-commande est traduite par le calcul d'une matrice d'interaction [Cha90] exprimant la variation des primitives observées dans l'image en fonction du mouvement de la caméra. Cette matrice, assimilable à un jacobien, dépend des observations dans l'image et des caractéristiques 3D de la scène. Elle n'est pas nécessairement refaire l'estimation à chaque prise de vue mais peut être, par exemple, calculée à la position de référence désirée. Souvent seule l'information de profondeur, fortement entachée de bruit, peut être à nouveau estimée ce qui alors ne nécessite qu'une estimation partielle de la pose [Gan99].

Considérons la valeur courante \mathbf{s}_i des informations visuelles observées par la caméra, représentant la configuration d'une primitive extraite dans l'image, et \mathbf{s}_i^* la valeur de référence

de \mathbf{s}_i à atteindre. Prenons à présent l'exemple d'un point de coordonnées $\mathbf{s}_i = [u_i v_i]^T$ en pixels dans le plan image. La matrice d'interaction $\mathbf{L}_{\mathbf{s}_i}$ associée à \mathbf{s}_i , est définie par la relation liant la dérivée des coordonnées du point dans l'image, au torseur cinématique $\mathbf{v} = [\nu^T \omega^T]^T$ de la caméra :

$$\dot{\mathbf{s}}_i = \frac{\delta \mathbf{s}_i}{\delta \mathbf{r}}(\mathbf{s}_i, Z_i, \mathbf{K}) \mathbf{v} = \mathbf{L}_{\mathbf{s}_i}(\mathbf{s}_i, Z_i, \mathbf{K}) \mathbf{v} \quad (1.28)$$

où \mathbf{K} représente la matrice des paramètres intrinsèques (voir équation (1.5)), Z_i désigne la profondeur du point 3D correspondant à \mathbf{s}_i , et \mathbf{r} définit le vecteur représentant la situation de l'effecteur du robot dans l'espace cartésien.

On a :

$$\mathbf{L}_{\mathbf{s}_i}(\mathbf{s}_i, Z_i, \mathbf{K}) = \begin{pmatrix} \alpha_u & \alpha_{uv} \\ 0 & \alpha_v \end{pmatrix} \begin{pmatrix} -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i y_i & -(1+x_i^2) & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & (1+y_i^2) & -x_i y_i & -x_i \end{pmatrix} \quad (1.29)$$

où les coordonnées métriques x_i et y_i sont fonctions des coordonnées pixels u_i et v_i et des paramètres intrinsèques de la caméra par :

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_u} & -\frac{\alpha_{uv}}{\alpha_u \alpha_v} \\ 0 & \frac{1}{\alpha_v} \end{pmatrix} \begin{pmatrix} u_i - u_0 \\ v_i - v_0 \end{pmatrix} \quad (1.30)$$

La fonction de tâche \mathbf{e} correspondant à la régulation de $\mathbf{s} - \mathbf{s}^*$, avec $\mathbf{s} = (\mathbf{s}_1^T \mathbf{s}_2^T \dots \mathbf{s}_p^T)^T$ correspondant au vecteur contenant les informations visuelles de p points, s'écrit :

$$\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*) \quad (1.31)$$

où $\mathbf{C} = \frac{\delta \mathbf{e}}{\delta \mathbf{s}}$ peut être choisi égale à la pseudo-inverse de la matrice d'interaction calculée pour $\mathbf{s} = \mathbf{s}^*$ et $\mathbf{z} = \mathbf{z}^*$ [Cha90, ECFP92], où \mathbf{z} est le vecteur courant de la profondeur des points, et \mathbf{z}^* correspond au vecteur \mathbf{z} à la position désirée. Comme en général les valeurs de \mathbf{K} et \mathbf{z}^* sont inconnues, on choisit :

$$\mathbf{C} = \widehat{\mathbf{L}}_s^+(\mathbf{s}^*, \widehat{\mathbf{z}}^*, \widehat{\mathbf{K}}) \quad (1.32)$$

On notera qu'un minimum de quatre points est nécessaire pour réaliser une tâche de positionnement rigide, et donc pour avoir une fonction de tâche admissible. En effet, avec trois points le système peut rencontrer une singularité de la matrice d'interaction [Mic93].

1.4.3.3 Commande dans l'espace 2D

La cible est supposée statique, et la matrice de transformation repère caméra/repère effecteur connue. En imposant une convergence exponentielle découplée de la fonction de tâche, la loi de commande en vitesse de la caméra est alors :

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}^{-1} \hat{\mathbf{e}} \quad (1.33)$$

où λ est un scalaire positif permettant de modifier la vitesse de convergence, et $\widehat{\mathbf{L}} = \mathbf{C} \widehat{\mathbf{L}}_s$.

1.5 Conclusion

Ce premier chapitre a permis de présenter différents domaines de recherche de la vision par ordinateur, que nous retrouvons dans notre approche d'acquisition des images et de reconstruction 3D. Chacun de ces domaines a fait l'objet de nombreux travaux de recherche, ce que nous proposons à présent c'est une nouvelle approche de reconstruction, intégrant des méthodes existantes performantes, pour une application spécifique sur des images sous-marines.

Un certain nombre de méthodes d'extraction et d'appariement de points pouvant être appliquées à tout type d'images de scènes naturelles ont été présentées. Cependant elles n'ont pas toutes le même temps de calcul, la même précision ou niveau de robustesse. Ainsi, nous verrons par la suite que l'asservissement visuel et la reconstruction 3D, bien qu'ayant une approche semblable au niveau de l'extraction et de l'appariement des primitives, utiliseront des méthodes différentes adaptées aux contraintes du milieu, ou à la précision du résultat souhaitée, en fonction de la contrainte de temps.

Le suivi visuel quant à lui, possède un rôle essentiel lors d'un asservissement visuel, puisque de sa précision et de sa robustesse aux perturbations, dépend la convergence de la caméra vers la position de référence. Nous avons présenté brièvement quelques approches générales de suivi visuel, et plus particulièrement deux méthodes, qui sont la méthode KLT et l'ESM, utilisées par la suite pour nos expérimentations.

Suivant la tâche à accomplir (positionnement, suivi, saisie, etc) et les informations dont on dispose sur le réglage des caméras et sur la scène observée, il existe un nombre important de méthodes d'asservissement visuel. Les aspects importants à prendre en compte pour le choix d'une méthode d'asservissement visuel pour notre application, sont de permettre un contrôle direct dans l'image afin d'assurer la visibilité de la tâche, de découpler la loi de commande en rotation de celle en translation (ce qui permet l'analyse de stabilité de l'asservissement et de sa robustesse aux erreurs de calibration), et d'utiliser un nombre minimum d'informations a priori sur la structure de l'objet observé.

La méthode de reconstruction qui semble la mieux adaptée à notre problème est celle à partir de la géométrie épipolaire. En effet, la lumière structurée est plus indiquée sur des surfaces de texture uniforme, et il serait difficile de projeter un motif avec une lumière uniforme sur une scène naturelle sous-marine, étant donné les problèmes d'éclairage sous l'eau (problèmes déjà abordés en introduction). Les méthodes basées sur les silhouettes paraissent trop imprécises sur les parties concaves de l'objet, les méthodes à partir du mouvement sont plus sensibles au bruit étant donnée la proximité des images consécutives n'offrant qu'une faible différence de position entre les images, et celles basées sur le critère de photoconsistance sont trop dépendantes d'un éclairage uniforme sur toutes les poses. La prochaine partie mettra en avant la stratégie mise en œuvre pour la capture des images servant à faire la reconstruction 3D, ce qui constitue une partie de nos contributions.

Chapitre 2

L'acquisition des images

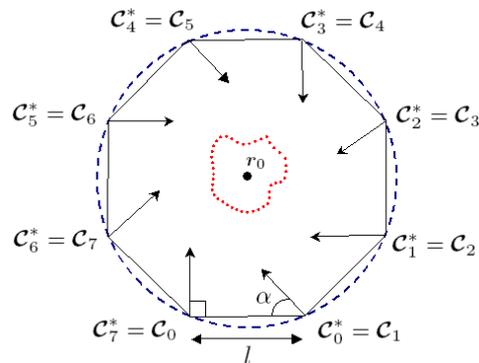
L'acquisition des images est la première étape essentielle de notre approche avant de faire la reconstruction 3D. La méthode d'acquisition développée ici, permet de simplifier la programmation de mission en mer d'acquisition d'images, et de connaître la position et l'orientation des prises de vue, dans le but de calculer la reconstruction 3D d'objets sous-marin. L'idée générale est de contraindre les prises de vue de façon à couvrir la surface à reconstruire avec un bon taux de recouvrement dans les images, et de contrôler précisément la position des prises de vue. Ainsi nous proposons de générer la trajectoire d'acquisition en fonction de la géométrie d'une tête stéréo, et de contrôler la précision d'acquisition en exécutant la trajectoire par asservissement visuel.

La méthode consistant à générer des trajectoires est détaillée dans la première partie, ainsi que sa mise en œuvre qui peut se faire soit par asservissement visuel, soit par une méthode alternative à la vision, moins flexible par rapport aux modifications de trajectoires, l'asservissement en position par programme robot. La deuxième partie est consacrée aux simulations du suivi de trajectoires par asservissement visuel. La troisième partie présente les résultats expérimentaux. Cette dernière partie décrit les évolutions du système d'acquisition depuis les essais dans l'air, jusqu'aux expérimentations en mer, en passant par les phases de validation en bassin.

Il s'agit donc de réaliser tout d'abord l'acquisition des images par asservissement visuel à l'aide de deux caméras sous-marines, puis de faire le calcul de reconstruction 3D à partir de ces images, ce qui sera l'objet du prochain chapitre.

2.1 Principe de la méthode d'acquisition par génération de trajectoires

La méthode d'acquisition comporte deux aspects, la génération de trajectoires, et l'exécution de ces trajectoires. Dans cette partie nous nous intéressons à la mise en équation du déplacement d'une tête stéréo sur une trajectoire ne dépendant que de la géométrie entre deux caméras.



- - - - Trajectory of the cameras

..... Object

FIG. 2.1 – Trajectoires induites par la géométrie de la tête stéréo

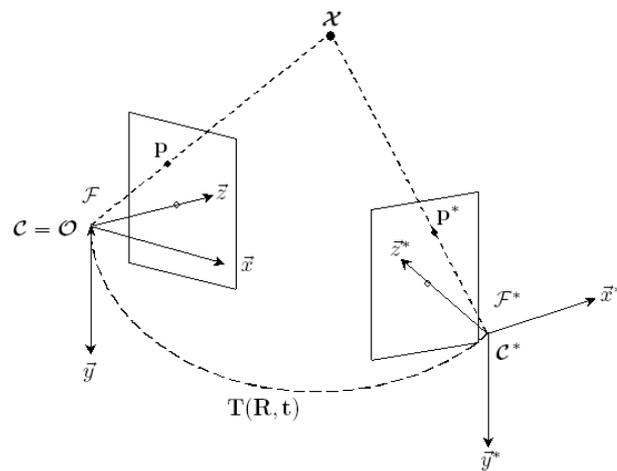


FIG. 2.2 – Projection d'un point \mathcal{X} de l'espace dans l'image de deux caméras

La géométrie d'un banc stéréo est caractérisée par la transformation (rotation, translation) liant les deux caméras. Une fois la géométrie fixée, les déplacements successifs d'une caméra sur la position de la seconde caméra décrivent une trajectoire unique (cf. Figure 2.1). Pour illustrer cette idée, considérons le cas où les axes optiques des deux caméras sont dans le même plan. L'angle α et l'écartement l vont déterminer la matrice de transformation, composée d'une rotation et d'une translation, entre les repères des deux

caméras (cf. Figure 2.2). Nous allons montrer que, dans le cas général, l'origine des repères des caméras se déplace nécessairement sur la surface d'un cylindre sous la forme d'une hélice, d'un cercle, ou d'une droite (cercle de rayon infini) à condition de ne pas modifier la géométrie de la tête [BMR⁺06]. Le but de cette mise en équation de la trajectoire des caméras en fonction de la géométrie de la tête stéréo, est de pouvoir déduire le choix du réglage des caméras en fonction des dimensions de l'objet à reconstruire. Ceci a fait l'objet d'une spécification de la géométrie de la tête stéréo, qui permet de calculer la taille maximale de l'objet à reconstruire en fonction des paramètres de la transformation géométrique entre les caméras, ainsi que du réglage de leur focale. Cette étude tient compte du recouvrement des champs de vision des caméras, afin de toujours pouvoir observer une même zone de l'objet dans les deux caméras. L'espace de travail du bras manipulateur étant limité, le robot devra être positionné en différentes stations autour de l'objet si l'on souhaite en avoir une reconstruction complète par recollement de reconstructions partielles.

La démonstration qui suit est basée sur l'intégration de déplacements infinitésimaux afin d'obtenir un déplacement discrétisé. Soit \mathbf{T} , un élément du groupe spécial Euclidien de dimension trois $SE(3)$, la matrice de transformation entre les positions des deux caméras (cf. Figure (2.2)) :

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \quad \mathbf{T} \in SE(3) \quad (2.1)$$

où \mathbf{R} est la matrice de rotation (3×3), et \mathbf{t} est le vecteur de translation (3×1). La matrice de rotation peut être écrite comme une fonction de α :

$$\mathbf{R}(\alpha) = e^{[\mathbf{r}(\alpha)]_{\times}} \quad (2.2)$$

où \mathbf{r} est le vecteur de rotation correspondant à un angle $\alpha \in \mathbb{R}$ autour d'un axe fixe spécifié par le vecteur $\boldsymbol{\delta}$:

$$\mathbf{r}(\alpha) = \frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|} \alpha \quad (2.3)$$

avec :

$$\boldsymbol{\delta} = \begin{pmatrix} \delta_{tilt} \\ \delta_{pan} \\ 0 \end{pmatrix} \quad (2.4)$$

De même, le vecteur de translation dépend de α et de l'écartement \mathbf{l} entre les caméras :

$$\mathbf{t}(\alpha, \mathbf{l}) = -\mathbf{R}(\alpha) \mathbf{l} = -e^{[\mathbf{r}(\alpha)]_{\times}} \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \quad (2.5)$$

\mathbf{T} peut s'écrire :

$$\mathbf{T} = e^{\mathbf{A}} \quad \mathbf{A} \in se(3) \quad (2.6)$$

où $se(3)$ est l'algèbre de Lie associée au groupe de Lie, et :

$$\mathbf{A} = \begin{pmatrix} [\boldsymbol{\omega}]_{\times} & \boldsymbol{\nu} \\ \mathbf{0} & 0 \end{pmatrix} \quad (2.7)$$

où $\boldsymbol{\omega}$ est le vecteur de vitesse en rotation (3×1) tel que $\|\boldsymbol{\omega}\| < \pi$, et $\boldsymbol{\nu}$ est le vecteur de vitesse en translation (3×1). Soit $[\boldsymbol{\omega}]_{\times}$ la matrice antisymétrique (3×3) associée au vecteur $\boldsymbol{\omega}$.

Un déplacement infinitésimal de l'origine de l'image \mathcal{O} , laquelle coïncide avec le centre de projection \mathcal{C} , sur la trajectoire en fonction du temps t normalisé entre 0 et 1, est défini par :

$$\dot{\mathcal{C}}(t) = [\boldsymbol{\omega}]_{\times} \mathcal{C}(t) + \boldsymbol{\nu} \quad (2.8)$$

2.1.1 Cas mono-dimensionnel

Étudions tout d'abord la solution de l'équation différentielle du premier ordre pour des variables à une dimension, afin de simplifier par la suite le cas multi-dimensionnel :

$$\begin{cases} \dot{x}(t) = ax(t) + b \\ x(0) = 0 \end{cases}$$

Cette démonstration utilise les développements en série de Taylor.

$$\begin{aligned} x(t) &= -\left(\frac{1-e^{at}}{a}\right)b = -\left(\frac{1-\sum_{k=0}^{+\infty} \frac{(at)^k}{k!}}{a}\right)b \\ &= -\left(\frac{1-(1+(at)+\frac{1}{2!}(at)^2+\dots+\frac{1}{k!}(at)^k)}{a}\right)b \\ &= -\left(-1 - \frac{1}{2!}(at) - \frac{1}{3!}(at)^2 - \dots - \frac{1}{(k+2)!}(at)^{k+1}\right)bt \\ &= \left(1 + \sum_{k=0}^{+\infty} \frac{(at)^{k+1}}{(k+2)!}\right)bt \end{aligned}$$

Il s'agit à présent de retrouver la même forme d'expression du déplacement du centre de projection de la caméra dans le cas multi-dimensionnel.

2.1.2 Cas multi-dimensionnel

Le cas multi-dimensionnel correspondant un déplacement dans l'espace cartésien. Soient les changements de variables :

$$\boldsymbol{\Omega} = [\boldsymbol{\omega}]_{\times} \quad \mathbf{D} = \mathbf{T}^{-1}\boldsymbol{\Omega}\mathbf{T} \quad \mathbf{c} = \mathbf{T}^{-1}\boldsymbol{\nu} \quad \mathbf{z}(t) = \mathbf{T}^{-1}\mathcal{C}(t)$$

L'équation $\dot{\mathcal{C}}(t)$ devient donc :

$$\dot{\mathbf{z}}(t) = \mathbf{D}\mathbf{z}(t) + \mathbf{c}$$

$$\begin{cases} z_1(t) = -\left(\frac{1-e^{d_1 t}}{d_1}\right)c_1 = \alpha_1(t)c_1 \\ z_2(t) = -\left(\frac{1-e^{d_2 t}}{d_2}\right)c_2 = \alpha_2(t)c_2 \\ z_3(t) = -\left(\frac{1-e^{d_3 t}}{d_3}\right)c_3 = \alpha_3(t)c_3 \end{cases}$$

$$\mathbf{z} = \begin{pmatrix} \alpha_1(t) & 0 & 0 \\ 0 & \alpha_2(t) & 0 \\ 0 & 0 & \alpha_3(t) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

$$\mathcal{C}(t) = \mathbf{T} \begin{pmatrix} \alpha_1(t) & 0 & 0 \\ 0 & \alpha_2(t) & 0 \\ 0 & 0 & \alpha_3(t) \end{pmatrix} \mathbf{T}^{-1} \boldsymbol{\nu}$$

où :

$$\begin{pmatrix} \alpha_1(t) & 0 & 0 \\ 0 & \alpha_2(t) & 0 \\ 0 & 0 & \alpha_3(t) \end{pmatrix} = \left(\mathbf{I} + \frac{1}{2!} \mathbf{D}t + \frac{1}{3!} (\mathbf{D}t)^2 + \dots + \frac{1}{(k+2)!} (\mathbf{D}t)^{k+1} \right) t$$

$$= \left(\mathbf{I} + \sum_{k=0}^{+\infty} \frac{1}{(k+2)!} (\mathbf{D}t)^{k+1} \right) t$$

Ainsi l'expression devient :

$$\mathcal{C}(t) = \left(\mathbf{I} + \sum_{k=0}^{+\infty} \frac{1}{(k+2)!} \mathbf{T} (\mathbf{D}t)^{k+1} \mathbf{T}^{-1} \right) \boldsymbol{\nu} t = \left(\mathbf{I} + \sum_{k=0}^{+\infty} \frac{1}{(k+2)!} (\mathbf{T} \mathbf{D} \mathbf{T}^{-1} t)^{k+1} \right) \boldsymbol{\nu} t$$

$$\mathcal{C}(t) = \left(\mathbf{I} + \sum_{k=0}^{+\infty} \frac{1}{(k+2)!} (\boldsymbol{\Omega} t)^{k+1} \right) \boldsymbol{\nu} t$$

$$\mathcal{C}(t) = \left(\mathbf{I} + \frac{1}{2!} \boldsymbol{\Omega} t + \frac{1}{3!} (\boldsymbol{\Omega} t)^2 + \frac{1}{4!} (\boldsymbol{\Omega} t)^3 + \dots + \frac{1}{(k+2)!} (\boldsymbol{\Omega} t)^{k+1} \right) \boldsymbol{\nu} t$$

D'après le théorème de Cayley-Hamilton, le polynôme caractéristique d'une matrice 3×3 comme $\boldsymbol{\Omega}$ est :

$$\boldsymbol{\Omega}^3 - \text{trace}(\boldsymbol{\Omega}) \boldsymbol{\Omega}^2 + \frac{1}{2} (\text{trace}^2(\boldsymbol{\Omega}) - \text{trace}(\boldsymbol{\Omega}^2)) \boldsymbol{\Omega} - \det(\boldsymbol{\Omega}) \mathbf{I}_3 = 0$$

$$\text{Où } \frac{1}{2} (\text{Trace}^2(\boldsymbol{\Omega}) - \text{Trace}(\boldsymbol{\Omega}^2)) = \|\boldsymbol{\Omega}\|^2 = \|\boldsymbol{\omega}\|^2$$

Pour une matrice antisymétrique 3×3 : $\text{trace}(\boldsymbol{\Omega}) = 0$ et $\det(\boldsymbol{\Omega}) = 0$

Ainsi nous avons :

$$\boldsymbol{\Omega}^3 + \|\boldsymbol{\omega}\|^2 \boldsymbol{\Omega} = 0$$

Donc :

$$\boldsymbol{\Omega}^3 = -\|\boldsymbol{\omega}\|^2 \boldsymbol{\Omega}$$

$$\boldsymbol{\Omega}^4 = -\|\boldsymbol{\omega}\|^2 \boldsymbol{\Omega}^2$$

$$\boldsymbol{\Omega}^5 = -\|\boldsymbol{\omega}\|^2 \boldsymbol{\Omega}^3 = \|\boldsymbol{\omega}\|^4 \boldsymbol{\Omega}$$

$$\boldsymbol{\Omega}^6 = -\|\boldsymbol{\omega}\|^2 \boldsymbol{\Omega}^4 = \|\boldsymbol{\omega}\|^4 \boldsymbol{\Omega}^2$$

$$\Omega^7 = -\|\omega\|^2 \Omega^5 = -\|\omega\|^6 \Omega$$

$$\Omega^8 = -\|\omega\|^2 \Omega^6 = -\|\omega\|^6 \Omega^2$$

...

$$\Omega^{2k+1} = \|\omega\|^{2k} (-1)^k \Omega$$

$$\Omega^{2k+2} = \|\omega\|^{2k} (-1)^k \Omega^2$$

L'équation peut donc s'écrire :

$$\mathcal{C}(t) = (\mathbf{I} + \frac{1}{2!} \Omega t + \frac{1}{3!} \Omega^2 t^2 - \dots + \frac{(-1)^k}{(2k+2)!} \Omega \|\omega\|^{2k} t^{2k+1} - \frac{(-1)^{k+1}}{(2k+3)!} \Omega^2 \|\omega\|^{2k} t^{2k+2}) \nu t$$

Puisque $\frac{\Omega}{\|\omega\|} = \left[\frac{\omega}{\|\omega\|} \right]_{\times}$, on obtient :

$$\begin{aligned} \mathcal{C}(t) = & (\mathbf{I} + \frac{1}{2!} \left[\frac{\omega}{\|\omega\|} \right]_{\times} \|\omega\| t + \frac{1}{3!} \left[\frac{\omega}{\|\omega\|} \right]_{\times}^2 (\|\omega\| t)^2 - \dots \\ & \dots + \frac{(-1)^k}{(2k+2)!} \left[\frac{\omega}{\|\omega\|} \right]_{\times} (\|\omega\| t)^{2k+1} - \frac{(-1)^{k+1}}{(2k+3)!} \left[\frac{\omega}{\|\omega\|} \right]_{\times}^2 (\|\omega\| t)^{2k+2}) \nu t \end{aligned}$$

Avec $\left[\frac{\omega}{\|\omega\|} \right]_{\times}$ et $\left[\frac{\omega}{\|\omega\|} \right]_{\times}^2$ factorisés, l'équation devient :

$$\mathcal{C}(t) = \left(\mathbf{I} + \left[\frac{\omega}{\|\omega\|} \right]_{\times} \sum_{k=0}^{+\infty} \frac{(-1)^k (\|\omega\| t)^{2k+1}}{(2k+2)!} - \left[\frac{\omega}{\|\omega\|} \right]_{\times}^2 \sum_{k=0}^{+\infty} \frac{(-1)^{k+1} (\|\omega\| t)^{2k+2}}{(2k+3)!} \right) \nu t$$

$$\mathcal{C}(t) = \left(\mathbf{I} + \frac{\|\omega\| t}{2} \operatorname{sinc}^2 \left(\frac{\|\omega\| t}{2} \right) \left[\frac{\omega}{\|\omega\|} \right]_{\times} - (\operatorname{sinc}(\|\omega\| t) - 1) \left[\frac{\omega}{\|\omega\|} \right]_{\times}^2 \right) \nu t$$

Finalement nous obtenons :

$$\mathcal{C}(t) = \left(\mathbf{I} + \left(\frac{1 - \cos(\|\omega t\|)}{\|\omega t\|} \right) \left[\frac{\omega t}{\|\omega t\|} \right]_{\times} + (1 - \operatorname{sinc}(\|\omega t\|)) \left[\frac{\omega t}{\|\omega t\|} \right]_{\times}^2 \right) \nu t$$

Posons : $\alpha(t) = \left(\frac{1 - \cos(\|\omega t\|)}{\|\omega t\|} \right)$ et $\beta(t) = (1 - \operatorname{sinc}(\|\omega t\|))$

$$x(t) = \left(\nu_x + \alpha(t) \left(\frac{-\nu_y \omega_z + \nu_z \omega_y}{\|\omega\|} \right) + \beta(t) \left(\frac{-\nu_x (\omega_y^2 + \omega_z^2) + \nu_y \omega_x \omega_y + \nu_z \omega_x \omega_z}{\|\omega\|^2} \right) \right) t$$

$$y(t) = \left(\nu_y + \alpha(t) \left(\frac{\nu_x \omega_z - \nu_z \omega_x}{\|\boldsymbol{\omega}\|} \right) + \beta(t) \left(\frac{\nu_x \omega_x \omega_y - \nu_y (\omega_z^2 + \omega_x^2) + \nu_z \omega_y \omega_z}{\|\boldsymbol{\omega}\|^2} \right) \right) t$$

$$z(t) = \left(\nu_z + \alpha(t) \left(\frac{-\nu_x \omega_y + \nu_y \omega_x}{\|\boldsymbol{\omega}\|} \right) + \beta(t) \left(\frac{\nu_x \omega_x \omega_z + \nu_y \omega_y \omega_z - \nu_z (\omega_y^2 + \omega_x^2)}{\|\boldsymbol{\omega}\|^2} \right) \right) t$$

Maintenant posons :

$$\boldsymbol{\gamma} = \begin{pmatrix} -\nu_y \omega_z + \nu_z \omega_y \\ \nu_x \omega_z - \nu_z \omega_x \\ -\nu_x \omega_y + \nu_y \omega_x \end{pmatrix} \quad \boldsymbol{\zeta} = \begin{pmatrix} -\nu_x (\omega_y^2 + \omega_z^2) + \nu_y \omega_x \omega_y + \nu_z \omega_x \omega_z \\ \nu_x \omega_x \omega_y - \nu_y (\omega_z^2 + \omega_x^2) + \nu_z \omega_y \omega_z \\ \nu_x \omega_x \omega_z + \nu_y \omega_y \omega_z - \nu_z (\omega_y^2 + \omega_x^2) \end{pmatrix}$$

L'équation de la trajectoire devient :

$$\begin{aligned} \mathcal{C}(t) &= \left(\boldsymbol{\nu} + \alpha(t) \left(\frac{\boldsymbol{\gamma}}{\|\boldsymbol{\omega}\|} \right) + \beta(t) \left(\frac{\boldsymbol{\zeta}}{\|\boldsymbol{\omega}\|^2} \right) \right) t \\ &= \boldsymbol{\nu} t + \left(\frac{1 - \cos(\|\boldsymbol{\omega}t\|)}{\|\boldsymbol{\omega}\|} \right) \left(\frac{\boldsymbol{\gamma}}{\|\boldsymbol{\omega}\|} \right) + \left(\frac{\|\boldsymbol{\omega}t\| - \sin(\|\boldsymbol{\omega}t\|)}{\|\boldsymbol{\omega}\|} \right) \left(\frac{\boldsymbol{\zeta}}{\|\boldsymbol{\omega}\|^2} \right) \\ &= \frac{\boldsymbol{\gamma}}{\|\boldsymbol{\omega}\|^2} + \cos(\|\boldsymbol{\omega}t\|) \left(-\frac{\boldsymbol{\gamma}}{\|\boldsymbol{\omega}\|^2} \right) + \sin(\|\boldsymbol{\omega}t\|) \left(-\frac{\boldsymbol{\zeta}}{\|\boldsymbol{\omega}\|^3} \right) + \left(\boldsymbol{\nu} + \frac{\boldsymbol{\zeta}}{\|\boldsymbol{\omega}\|^2} \right) t \end{aligned}$$

Supposons l'axe d'un cylindre passant par le point (X_0, Y_0, Z_0) , et orienté dans la direction du vecteur unitaire \boldsymbol{w} ; de plus nous supposons que \boldsymbol{u} et \boldsymbol{v} sont des vecteurs unitaires mutuellement perpendiculaires, et perpendiculaires à l'axe du cylindre. Les équations paramétriques de la surface d'un cylindre sont :

$$\left. \begin{aligned} x(t) &= X_0 + \cos(\theta)u_1 + \sin(\theta)v_1 + w_1t \\ y(t) &= Y_0 + \cos(\theta)u_2 + \sin(\theta)v_2 + w_2t \\ z(t) &= Z_0 + \cos(\theta)u_3 + \sin(\theta)v_3 + w_3t \end{aligned} \right\} \quad \mathcal{C}(t) = \boldsymbol{r}_0 + \cos(\theta)\boldsymbol{u} + \sin(\theta)\boldsymbol{v} + \boldsymbol{w}t$$

Nous pouvons déduire de ceci que :

$$\theta = \|\boldsymbol{\omega}t\|, \quad \boldsymbol{r}_0 = \frac{\boldsymbol{\gamma}}{\|\boldsymbol{\omega}\|^2},$$

$$\boldsymbol{u} = \left(-\frac{\boldsymbol{\gamma}}{\|\boldsymbol{\omega}\|^2} \right), \quad \boldsymbol{v} = \left(-\frac{\boldsymbol{\zeta}}{\|\boldsymbol{\omega}\|^3} \right), \quad \boldsymbol{w} = \left(\boldsymbol{\nu} + \frac{\boldsymbol{\zeta}}{\|\boldsymbol{\omega}\|^2} \right)$$

En conclusion si nous appliquons une vitesse constante de translation et de rotation à une caméra, la trajectoire décrite par celle-ci suit toujours la surface d'un cylindre.

Les trois vecteurs unitaires \boldsymbol{u} , \boldsymbol{v} et \boldsymbol{w} forment un trièdre direct. En incluant \boldsymbol{r}_0 , ils caractérisent le cylindre qui définit la trajectoire. Puisque \boldsymbol{u} , \boldsymbol{v} , \boldsymbol{w} et \boldsymbol{r}_0 dépendent des composantes des variables $\boldsymbol{\nu}$ et $\boldsymbol{\omega}$, lesquelles dépendent elles-même de la matrice de transformation $\mathbf{T}(\mathbf{R}, \mathbf{t})$ reliant la position des deux caméras (cf. Equations 2.1,2.6,2.7), nous pouvons affirmer que la géométrie de la tête stéréo induit la trajectoire. Parce que le point

\mathbf{r}_0 est situé sur l'axe du cylindre, si nous fixons la géométrie de la tête stéréo, nous pouvons également déterminer la distance à respecter entre les caméras et l'objet à étudier. Ainsi, il est possible de choisir la géométrie de la tête stéréo en fonction de la forme, du volume, et de l'orientation de l'objet sous-marin à reconstruire. Par exemple, pour une géométrie fixe de la tête stéréo, la trajectoire correspond à une ligne droite dans le cas où les caméras sont parallèles, un cercle si nous appliquons un angle pan, ou une hélice dans le cas d'un angle pan et tilt. Par exemple (cf. Figure (2.1) page 40)), un angle pan α et une distance l force la tête stéréo à décrire une trajectoire circulaire. Nous pouvons également composer des trajectoires plus complexes si la géométrie de la tête stéréo est modifiée durant l'asservissement visuel.

En conséquence, si une configuration géométrique de la tête stéréo correspond à une seule trajectoire, une même trajectoire peut être induite par des combinaisons variables d'angles et d'écartements entre les caméras, lesquels ont un effet seulement sur le nombre de prises de vues le long de la trajectoire. Le nombre minimum d'images acquises pour faire la reconstruction 3D, dépendra de la géométrie de la tête stéréo permettant d'assurer un recouvrement des champs de vision des caméras suffisant pour réaliser un asservissement visuel. Concernant le nombre de vues nécessaires à la reconstruction 3D, nous pouvons mentionner les recherches menées par [TFZ98, RP05], lesquelles rendent possible la sélection d'un sous-ensemble approprié d'images de la séquence afin d'estimer la géométrie épipolaire entre les vues, puis de calculer la reconstruction 3D. Mais cette sélection se fait en post-traitement une fois qu'une séquence d'images ait déjà été acquise.

2.2 Simulations du suivi de trajectoire par asservissement visuel

Les résultats de simulation ont permis de tester la chaîne complète d'asservissement visuel en prenant en compte les données géométriques et cinématiques réelles du bras (butées articulaires, singularités, vitesses articulaires maximales, dimensions des axes), et des caméras (dimensions du capteur, taille des pixels et longueur de la focale) (cf. Figure 2.3). Les simulations ont été réalisées pour différentes valeurs de l'écartement et de l'angle entre les caméras, en respectant la distance nécessaire entre les caméras et l'objet à reconstruire (distance définie en considérant l'équation de la trajectoire en fonction de la géométrie de la tête stéréo, et le champ de vision des caméras). Des estimations des paramètres intrinsèques et des profondeurs ont été prises en compte, et du bruit a été rajouté sur les projections des points 3D représentant la cible des caméras. La trajectoire réalisée est comparée avec la trajectoire théorique pour contrôler la dérive des caméras due au bruit et aux estimations. La simulation de l'asservissement visuel du bras développée en début de thèse est un outil important de comparaison avec les expérimentations. Il permet également de planifier les positions du bras, et le nombre de déplacements de la tête stéréo lors des expérimentations en fonction de l'espace de travail du bras manipulateur choisi, et de la configuration des caméras.

La figure (2.4) représente une simulation de la trajectoire de la tête stéréo portée par l'effecteur du bras manipulateur du robot sous-marin. Dans la figure (2.4), la caméra *Cam1* est montée sur pan & tilt pour pouvoir régler l'angle entre les caméras; elle est utilisée pour l'acquisition des images de référence. La caméra *Cam2* est fixe, et elle est commandée durant l'asservissement visuel. Les deux caméras ont des paramètres intrinsèques différents. Dans ces figures, le nuage de points correspond à un objet 3D simulant la forme d'une cheminée hydrothermale. La trajectoire des caméras (et par conséquent la géométrie de la tête stéréo) est choisie suivant la taille et la forme de l'objet étudié. Dans cet exemple, la trajectoire est induite par une géométrie réglée avec une distance inter-axes des caméras de 0,3 mètre, et un angle pan de 18 degrés. Les figures (2.4)(a) et (2.4)(b) montrent une vue de côté et de dessus de la trajectoire générée par simulation. La trajectoire est représentée par les positions successives de la caméra de référence *Cam1*. Elle est ici composée de trois quarts de cercles espacés en hauteur de la largeur de la tête stéréo. La première position du bras pour chaque quart de cercle est calculée afin d'utiliser au maximum l'espace de travail du bras, pour faire la plus grande trajectoire possible. Le bras est positionné en début de trajectoire par programme robot. Le véhicule sous-marin étant posé au sol lors de la procédure d'acquisition des images, il n'est pas possible de faire directement le tour de l'objet à reconstruire. Il est donc nécessaire de déplacer le robot autour de l'objet, et de répéter à chaque fois la procédure d'acquisition.

2.2.1 Asservissement visuel 2D classique

Le but de l'asservissement visuel [ECFP92] est de contrôler l'organe effecteur d'un robot en utilisant les informations provenant de capteurs visuels. La plupart des techniques d'asservissement visuel sont basées sur l'apprentissage d'une position de référence par

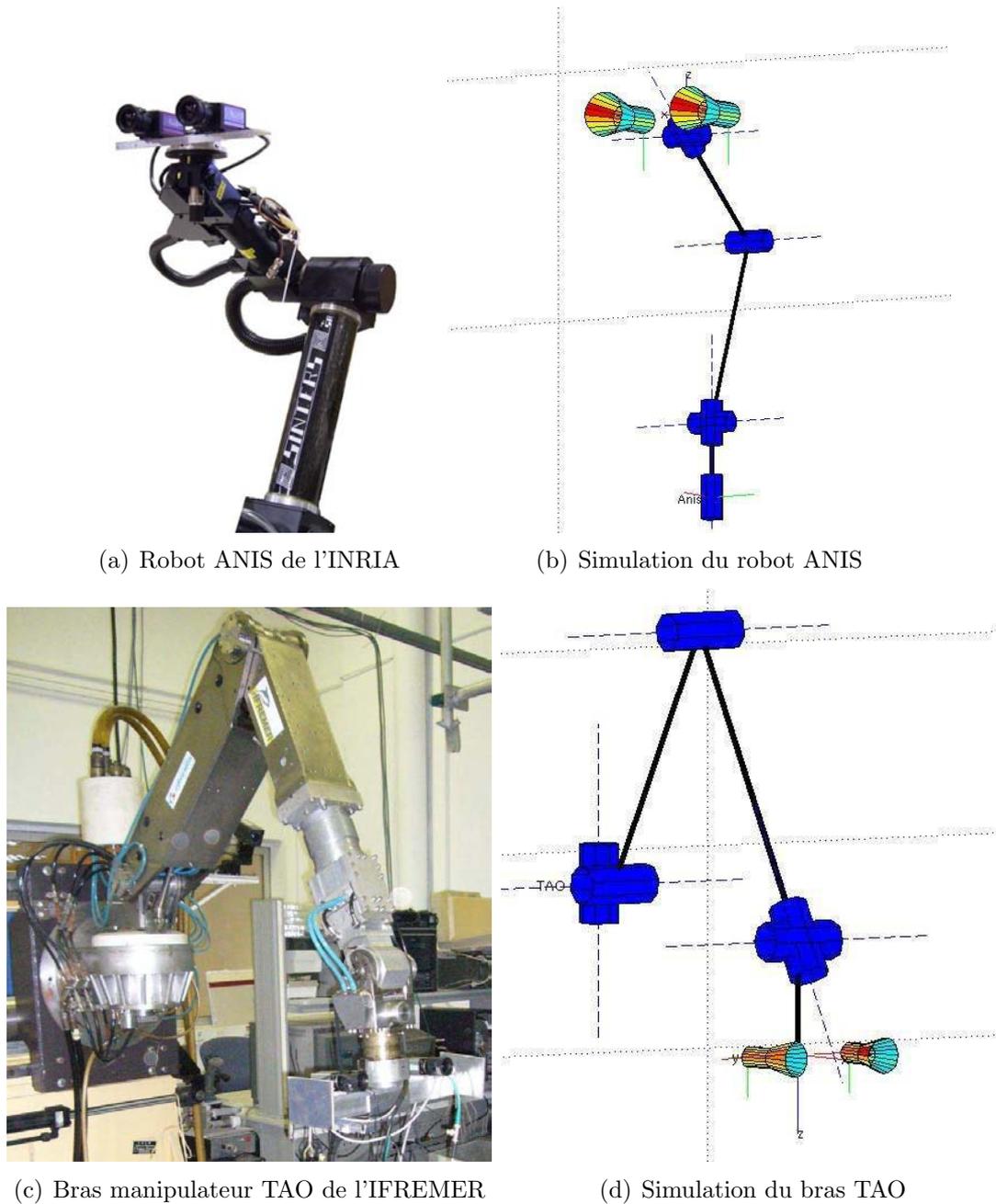


FIG. 2.3 – Description et modélisation des bras manipulateurs

l'observation, c'est la méthode dite « teaching-by-showing » [HHC96]. Il s'agit ensuite de mettre le système en bout de bras manipulateur dans la position de référence par rapport à l'objet observé. L'image de référence correspondant à la position désirée est ici acquise par la caméra droite de la tête stéréo (Fig. 2.2). La caméra gauche est donc la caméra asservie sur la position de référence. Une fois que la caméra gauche a convergé sur la position désirée, une nouvelle image de référence est acquise à partir de la nouvelle position de la caméra droite. Puis on recommence une nouvelle fois l'asservissement visuel et ainsi

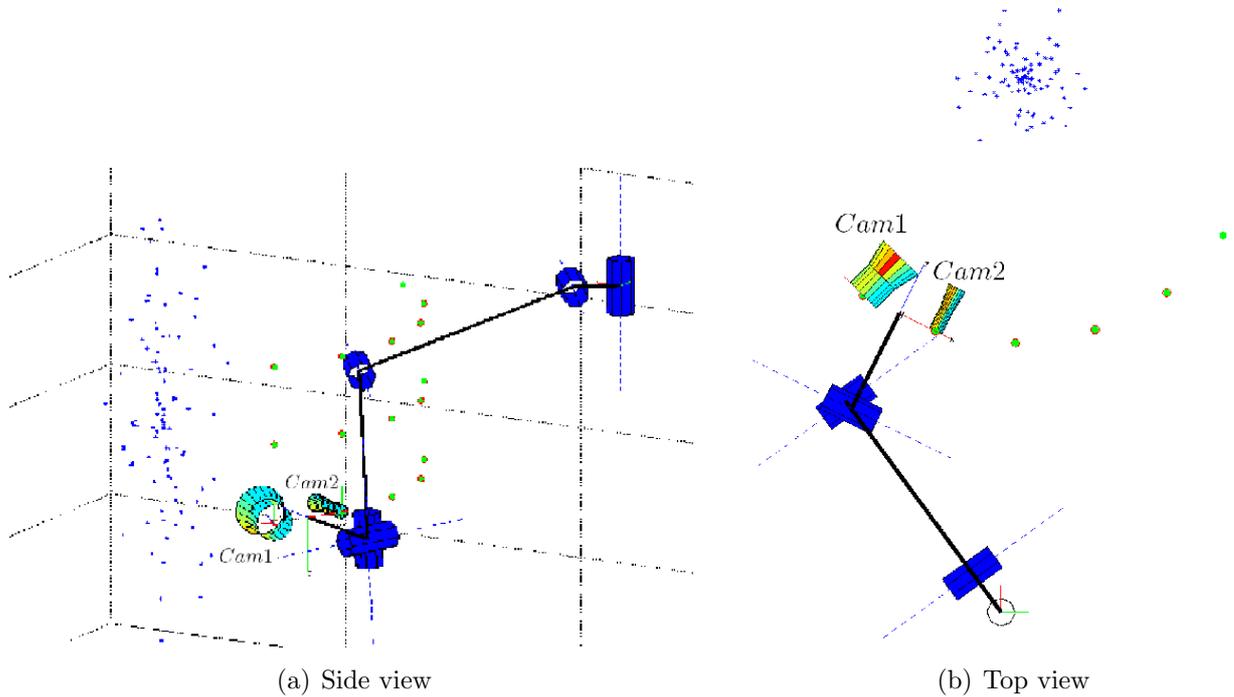
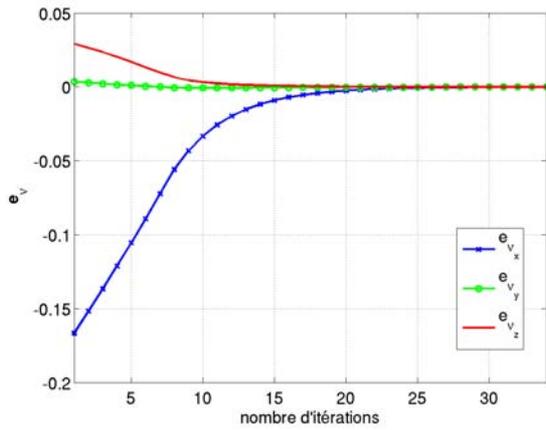


FIG. 2.4 – Simulation des trajectoires

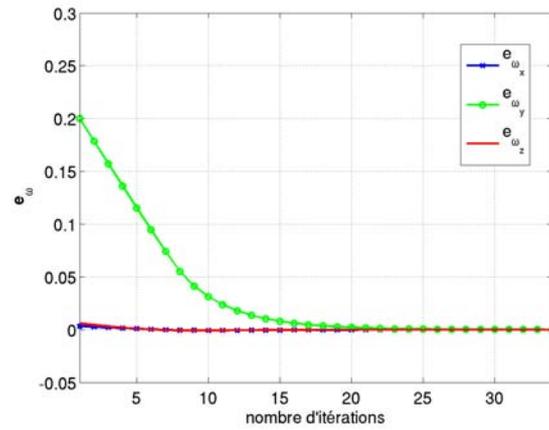
de suite, jusqu'à décrire la trajectoire désirée. La commande classique d'asservissement visuel 2D par fonction de tâche présentée en première partie, nécessite d'avoir des caméras calibrées afin de pouvoir faire converger les primitives extraites dans l'image courante vers celles de l'image de référence. La figure (2.5) représente la fonction de tâche calculée pour deux asservissements successifs, pour un écartement des caméras de 30cm, et un angle de 18° de la caméra droite par rapport à la caméra gauche. Les paramètres intrinsèques des deux caméras sont ici identiques. Si deux caméras différentes sont utilisées, il faut modifier les coordonnées des points de l'image de référence avec les paramètres intrinsèques des deux caméras, sinon la caméra asservie convergera sur une autre position comme illustré dans la figure (2.6). Chaque position de la caméra gauche est représentée par un rond vert plein, et celle de la caméra droite par un cercle rouge vide (Figure 2.6(c)). Pour un asservissement visuel réussi les cercles devraient se superposer comme sur la figure (2.4), ce qui n'est pas le cas.

2.2.2 Asservissement visuel invariant aux paramètres intrinsèques des caméras

Nous avons vu précédemment que la matrice de projection entre le repère monde et le repère image dépend entre autres des paramètres de calibration de la caméra. L'asservissement visuel repose généralement sur la connaissance *a priori* de ces paramètres, le contrôle de la caméra pouvant se faire dans l'espace cartésien ou directement dans l'image. Beaucoup d'approches ont été proposées pour résoudre le problème de la com-

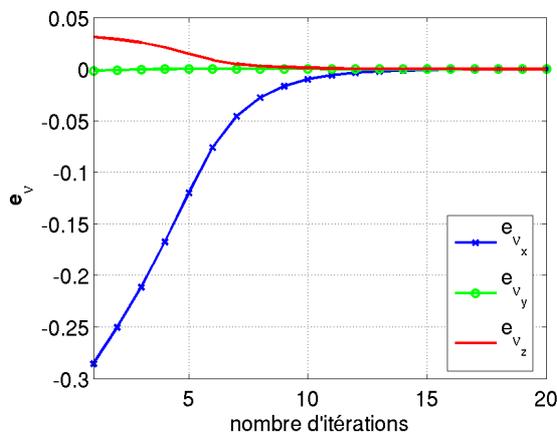


(a) Fonction de tâche : e_v

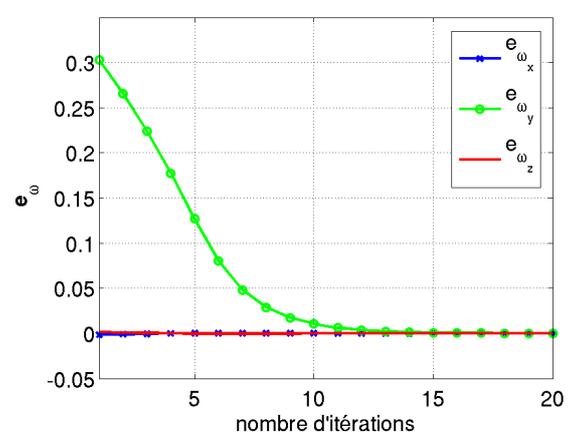


(b) Fonction de tâche : e_ω

Fonction de tâche du 1^{er} asservissement



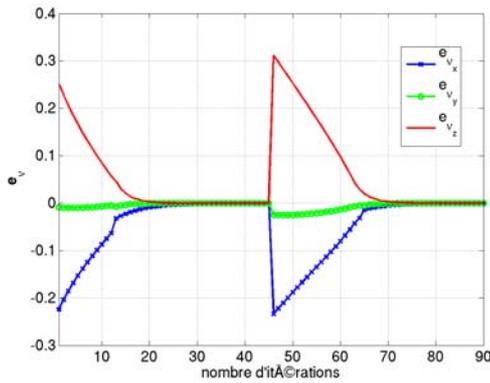
(c) Fonction de tâche : e_v



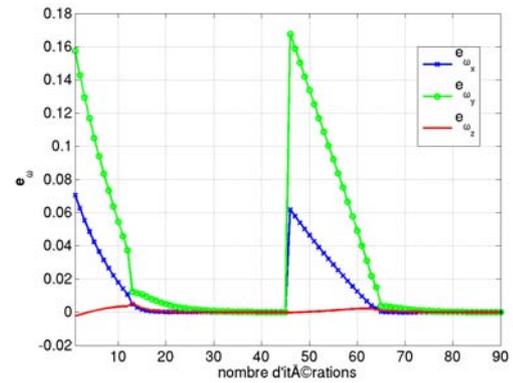
(d) Fonction de tâche : e_ω

Fonction de tâche du 2^{ième} asservissement

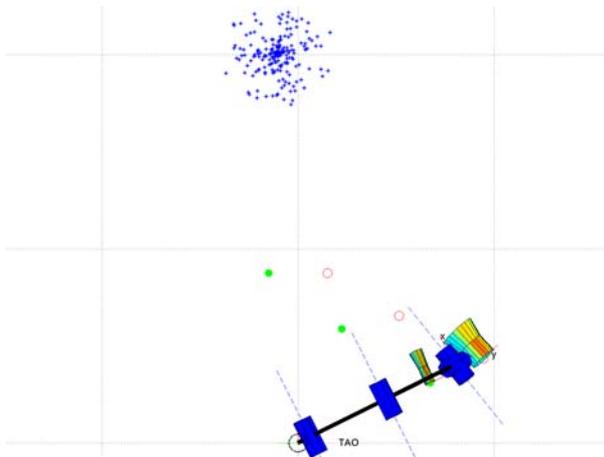
FIG. 2.5 – Commande 2D avec les caméras calibrées et sans perturbations



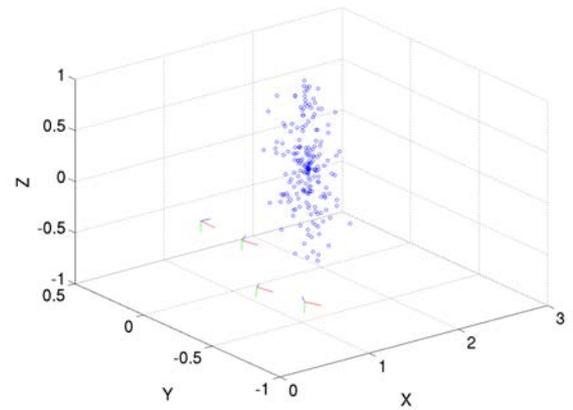
(a) Fonction de tâche : e_v



(b) Fonction de tâche : e_w



(c) Vue de dessus



(d) Repères des caméras au départ et après deux asservissements

FIG. 2.6 – Simulation d'une trajectoire par deux asservissements visuels successifs par la commande 2D, avec des paramètres intrinsèques différents entre l'image courante et l'image de référence.

mande basée sur la vision [CRE91, ECFP92, RCE89, SBE91]. La méthode que nous avons choisie pour calculer la commande a la particularité d'être indépendante des paramètres intrinsèques des caméras [Mal01a]. Le premier avantage pour notre système stéréo est de pouvoir utiliser deux caméras différentes. On peut donc apprendre l'image de référence avec la première caméra, puis réaliser l'asservissement avec la seconde caméra, et arriver à converger sur la même position bien que les paramètres des caméras soient différents. Le deuxième avantage est de pouvoir tester différents réglages des caméras pour réaliser l'asservissement visuel, sans avoir à recalibrer les caméras à chaque fois. Ceci nous permet de faire la calibration indispensable pour une reconstruction 3D métrique, seulement après une acquisition réussie de la séquence d'images. L'invariance aux paramètres intrinsèques est obtenue grâce à une transformation projective qui peut être calculée à partir des informations extraites dans l'image [Mal01b]. Le rôle de l'espace invariant où sont projetés les points, est d'éliminer la contrainte de calibration des paramètres intrinsèques (cf. §1.1.2.3 page 11), permettant ainsi d'augmenter considérablement les domaines d'application de l'asservissement visuel. Considérons n points de l'espace non colinéaires $\mathcal{X}_i = (X_i, Y_i, Z_i, 1)^\top \forall i \in 1, 2, \dots, n$, et deux caméras de centres de projection \mathcal{C} et \mathcal{C}^* coïncidant avec l'origine des repères des caméras (Fig. 2.2 page 40). Les points \mathcal{X}_i se projettent sur les images \mathcal{I} et \mathcal{I}^* en $\mathbf{m}_i = (x_i, y_i, 1)^\top$ et $\mathbf{m}_i^* = (x_i^*, y_i^*, 1)^\top$ suivant :

$$\zeta_i \mathbf{m}_i = [\mathbf{R} \ \mathbf{t}] \mathcal{X}_i, \quad \zeta_i^* \mathbf{m}_i^* = [\mathbf{I}_3 \ \mathbf{0}] \mathcal{X}_i \quad (2.9)$$

où ζ_i et ζ_i^* sont les profondeurs positives des points 3D, \mathbf{R} et \mathbf{t} sont respectivement la rotation et la translation entre les repères des deux caméras.

Les points \mathbf{m}_i et \mathbf{m}_i^* exprimés dans le repère image en pixels, correspondent aux points $\mathbf{p}_i = (u, v, 1)^\top$ et $\mathbf{p}_i^* = (u^*, v^*, 1)^\top$:

$$\mathbf{p}_i = \mathbf{K} \mathbf{m}_i, \quad \mathbf{p}_i^* = \mathbf{K}^* \mathbf{m}_i^* \quad (2.10)$$

où \mathbf{K} et \mathbf{K}^* sont respectivement les paramètres intrinsèques des caméras gauche et droite de la figure (2.2).

Afin de projeter respectivement les points \mathbf{p}_i et \mathbf{p}_i^* dans les espaces \mathcal{Q} et \mathcal{Q}^* invariants aux paramètres intrinsèques, une matrice symétrique (3×3) est calculée, dépendant exclusivement des points dans l'image :

$$\mathbf{S}_p = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i \mathbf{p}_i^\top, \quad \mathbf{S}_p^* = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i^* \mathbf{p}_i^{*\top} \quad (2.11)$$

Si les points observés ne sont pas colinéaires et $n > 3$, alors les matrices \mathbf{S}_p et \mathbf{S}_p^* sont des matrices symétriques définies positives, et peuvent s'écrire en utilisant la décomposition de Cholesky :

$$\mathbf{S}_p = \mathbf{T}_p \mathbf{T}_p^\top, \quad \mathbf{S}_p^* = \mathbf{T}_p^* \mathbf{T}_p^{*\top} \quad (2.12)$$

où \mathbf{T}_p et \mathbf{T}_p^* sont des matrices (3×3) triangulaires supérieures non singulières, qui nous permettent de calculer les points $\mathbf{q}_i \in \mathcal{Q}$ et $\mathbf{q}_i^* \in \mathcal{Q}^*$:

$$\mathbf{q}_i = \mathbf{T}_p^{-1} \mathbf{p}_i = (a_i, b_i, 1), \quad \mathbf{q}_i^* = \mathbf{T}_p^{*-1} \mathbf{p}_i^* = (a_i^*, b_i^*, 1) \quad (2.13)$$

Il est alors possible de définir une erreur, à partir des données extraites de l'image, qui ne dépend pas des paramètres intrinsèques de la caméra avec laquelle l'image a été prise. On peut donc contrôler le mouvement de la caméra et asservir l'erreur à zéro dans l'espace invariant, en utilisant une technique d'asservissement visuel basée sur l'approche par fonction de tâche. Les points \mathbf{q}_i et \mathbf{q}_i^* , projections dans l'espace invariant des points \mathbf{p}_i et \mathbf{p}_i^* de l'image courante et de l'image de référence d'après l'équation 2.13, sont contenus dans deux vecteurs : $\mathbf{s}^* = (\mathbf{q}_1^*, \mathbf{q}_2^*, \dots, \mathbf{q}_n^*)$ et $\mathbf{s} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n)$. La caméra gauche a convergé sur la position de référence quand $\mathbf{s} = \mathbf{s}^*$. La dérivée du vecteur \mathbf{s} s'écrit :

$$\dot{\mathbf{s}} = \mathbf{L}\mathbf{v} \quad (2.14)$$

où \mathbf{L} est la matrice d'interaction ($3n \times 6$), et \mathbf{v} le vecteur (6×1) représentant la vitesse cartésienne de la caméra asservie. La fonction de tâche est :

$$\mathbf{e} = \widehat{\mathbf{L}}^+(\mathbf{s} - \mathbf{s}^*) \quad (2.15)$$

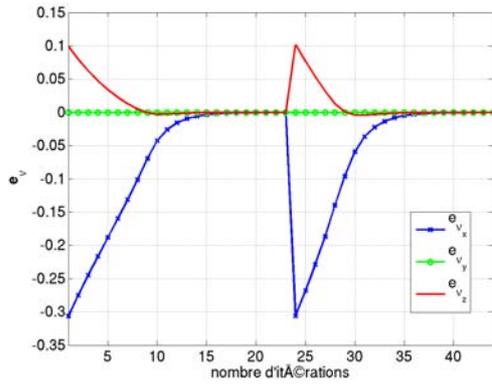
où $\widehat{\mathbf{L}}^+$ est une approximation de la pseudo-inverse de \mathbf{L} puisque la matrice des paramètres intrinsèques \mathbf{K} et les profondeurs des points ζ_i sont inconnues. Dans le but de contrôler le mouvement de la caméra en imposant la convergence exponentielle de la fonction de tâche, la loi de commande est :

$$\mathbf{v} = -\lambda\mathbf{e}, \quad \text{où } \lambda > 0 \quad (2.16)$$

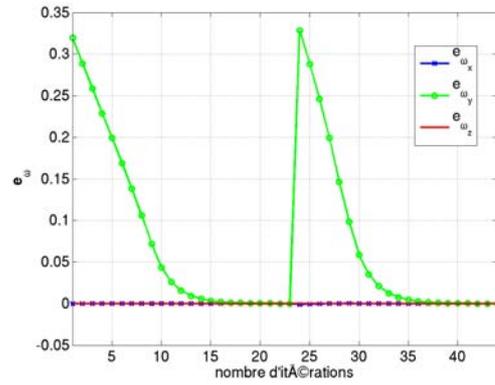
Cette commande a été testée avec un système stéréo, d'abord par simulation, puis en expérimentation comme nous le verrons par la suite. Comme cela a été montré dans [Mal01a], la loi de commande est stable même en présence d'erreurs de calibration.

La figure (2.7) représente la fonction de tâche calculée pour deux asservissements successifs avec la commande invariante, pour un écartement des caméras de 30cm, et un angle de 18° de la caméra droite par rapport à la caméra gauche. Les paramètres intrinsèques des deux caméras sont ici différents. Aucune modification des coordonnées des points de l'image de référence avec les vrais paramètres intrinsèques des deux caméras n'est nécessaire pour converger sur la bonne position comme le montre la figure (2.7.c). La figure (2.7.d) montre la trajectoire circulaire dans l'espace cartésien définie par les deux déplacements de la tête stéréo.

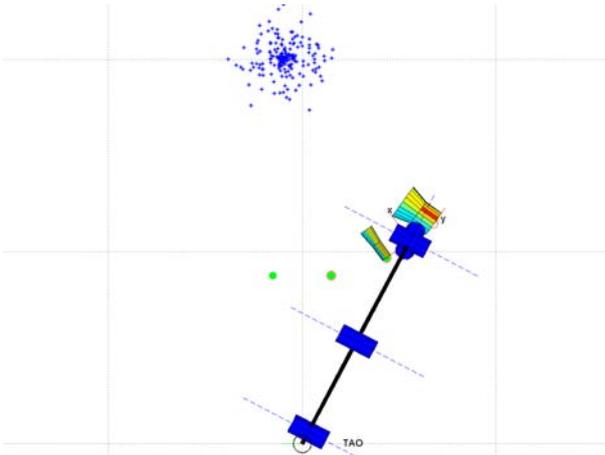
Les figures (2.8)(a) et (b) représentent le nuage de point vu par les deux caméras, à la fin de l'asservissement visuel une fois que la caméra asservie a convergé sur la position de référence. L'image de droite est l'image \mathcal{I}^* prise par la caméra de référence avec ses propres paramètres intrinsèques, tandis que l'image de gauche est l'image \mathcal{I} prise par la caméra asservie. On peut donc remarquer, étant donné que les paramètres intrinsèques des caméras ne sont pas les mêmes, que la position des points \mathbf{p}_i appartenant à \mathcal{I} ne correspond pas à la position des points \mathbf{p}_i^* appartenant à \mathcal{I}^* , alors même que la caméra a convergé sur la position de référence. A l'inverse, dans les figures (2.8)(c) et (d), les points \mathbf{q}_i et \mathbf{q}_i^* , qui représentent la projection des points $\mathbf{p}_i \in \mathcal{I}$ et $\mathbf{p}_i^* \in \mathcal{I}^*$ dans les espaces invariants aux paramètres intrinsèques \mathcal{Q} et \mathcal{Q}^* , sont à la même position. Ainsi, cette projection permet de calculer la loi de commande \mathbf{v} (cf. Equation 2.16 page 53) indépendante des paramètres des caméras, et de quantifier facilement la convergence des positions des caméras.



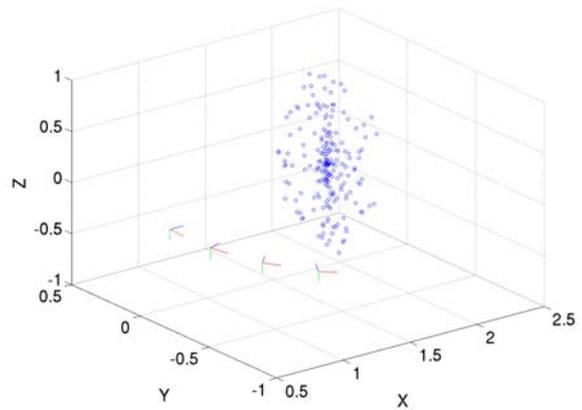
(a) Fonction de tâche : e_v



(b) Fonction de tâche : e_ω



(c) Vue de dessus



(d) Repères des caméras au départ et après deux asservissements

FIG. 2.7 – Simulation d'une trajectoire par deux asservissements visuels successifs avec des paramètres intrinsèques différents et la commande invariante.

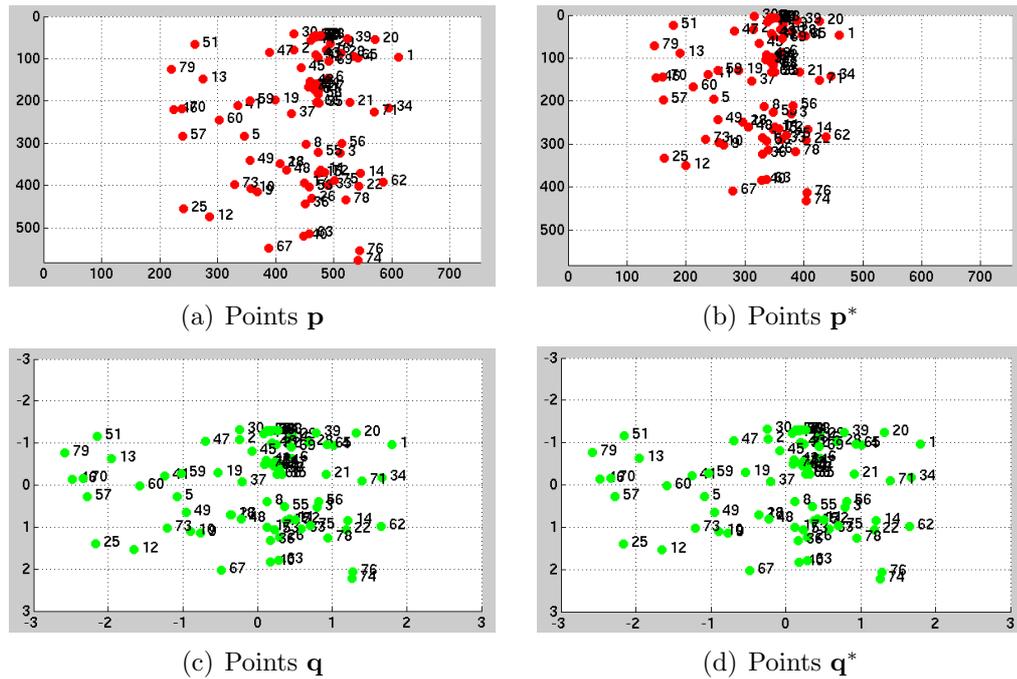


FIG. 2.8 – Projection d'un nuage de points vu par deux caméras stéréo ((a) et (b)) et reprojecté dans l'espace invariant ((c) et (d))

2.2.3 Simulations en présence d'erreurs et de perturbations

Différentes trajectoires adaptées à des objets de formes variées (cylindre, sphère, plusieurs plans, ...) ont été simulées. Du bruit sur les coordonnées des points a été rajouté pour simuler le bruit de la chaîne d'acquisition. L'objectif de l'utilisation des simulations est de tester la robustesse de la loi de commande, et d'observer le comportement du robot dans le but de préparer les expérimentations, et ainsi d'éviter des dommages sur le bras réel. De nombreux facteurs, liés à la robustesse des algorithmes employés, peuvent venir perturber le bon déroulement de l'asservissement visuel. C'est pourquoi il convient de les identifier et de les analyser afin de limiter leur influence sur la commande du robot.

– **Mauvais appariements de points :**

En cas de mauvais appariements, la caméra asservie va converger vers une autre position que la position de référence, s'il existe un minimum pour la fonction de tâche et si l'espace de travail du bras le permet. Un mauvais appariement provoque une mauvaise estimation de la profondeur du point 3D correspondant, ce qui a pour conséquence de modifier la commande de façon plus ou moins importante en fonction de l'importance de l'erreur, jusqu'à pouvoir la mettre en échec.

– **Dérive des points :**

En fonction de la robustesse de l'algorithme de suivi, et des informations qui caractérisent le point d'intérêt, celui-ci peut dériver avec le mouvement 3D de la caméra

(entraînant des déformations géométriques), si son voisinage représente un motif qui se répète dans l'image, ou si par exemple le point se situe sur le bord d'un objet et non un coin. Pour simuler la dérive de points suivis dans l'image on décale aléatoirement quelques points 3D faisant partie de la cible observée. Ceux-ci sont ensuite projetés dans l'image de la caméra gauche. Les points de l'image courante ne correspondant plus à la projection de la cible initiale, la caméra ne converge pas vers la bonne position. Une dérive trop importante des points peut provoquer des discontinuités importantes au niveau de la commande du bras. La perte ou l'élimination des points qui ont dérivé, peut entraîner une nouvelle discontinuité de la commande, mais surtout rétablit une convergence vers la bonne position. La dérive des points dépend de la robustesse de la méthode de suivi.

– **Bruits de la chaîne d'acquisition :**

Le bruit de la chaîne d'acquisition est simulé en rajoutant un bruit aléatoire issu d'une distribution normale (moyenne nulle et variance unité) sur les coordonnées en pixels des points projetés dans les images. Suivant son intensité, ce bruit va perturber le transitoire de la commande, puis la caméra va se mettre à osciller plus ou moins autour de la position de référence (Figure 2.9).

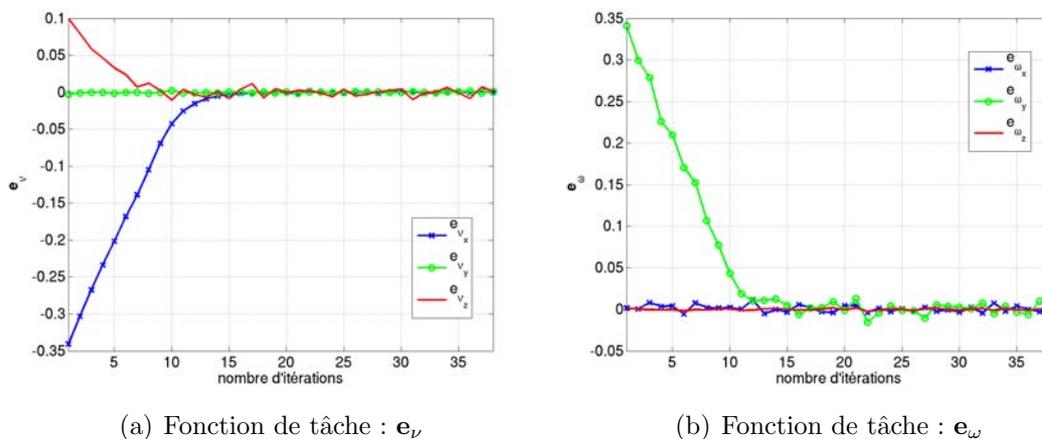


FIG. 2.9 – Simulation du comportement de la fonction de tâche dans l'espace invariant en présence de bruit sur les coordonnées des points dans l'image

– **Mauvaise estimation des profondeurs :**

L'estimation de la profondeur des points 3D intervient dans le calcul de la commande, au niveau de la matrice d'interaction. L'estimation des profondeurs, tout comme celle de la matrice d'interaction, n'est calculée qu'une seule fois au début de l'asservissement visuel, celle-ci n'influant que sur la forme du transitoire (pas sur la convergence exponentielle vers la position de référence). Ainsi on gagne en temps de calcul, l'estimation de la pseudoinverse de la matrice d'interaction prenant un temps important dans le processus d'asservissement exécuté en temps réel. Les profondeurs sont connues en simulation, alors qu'elles sont calculées par triangulation en expérimentation à partir des paramètres intrinsèques des caméras et de la géométrie de la

tête stéréo. Celles-ci n'interviennent dans le calcul de la commande qu'à un facteur près, d'où la nécessité d'une simple estimation. L'important pour réussir à converger sur la bonne position, est de respecter l'arrangement des points 3D, ainsi il ne faut pas modifier l'ordre des points vu depuis la caméra.

Sur la figure (2.10)) on peut voir que la fonction de tâche calculée dans l'espace

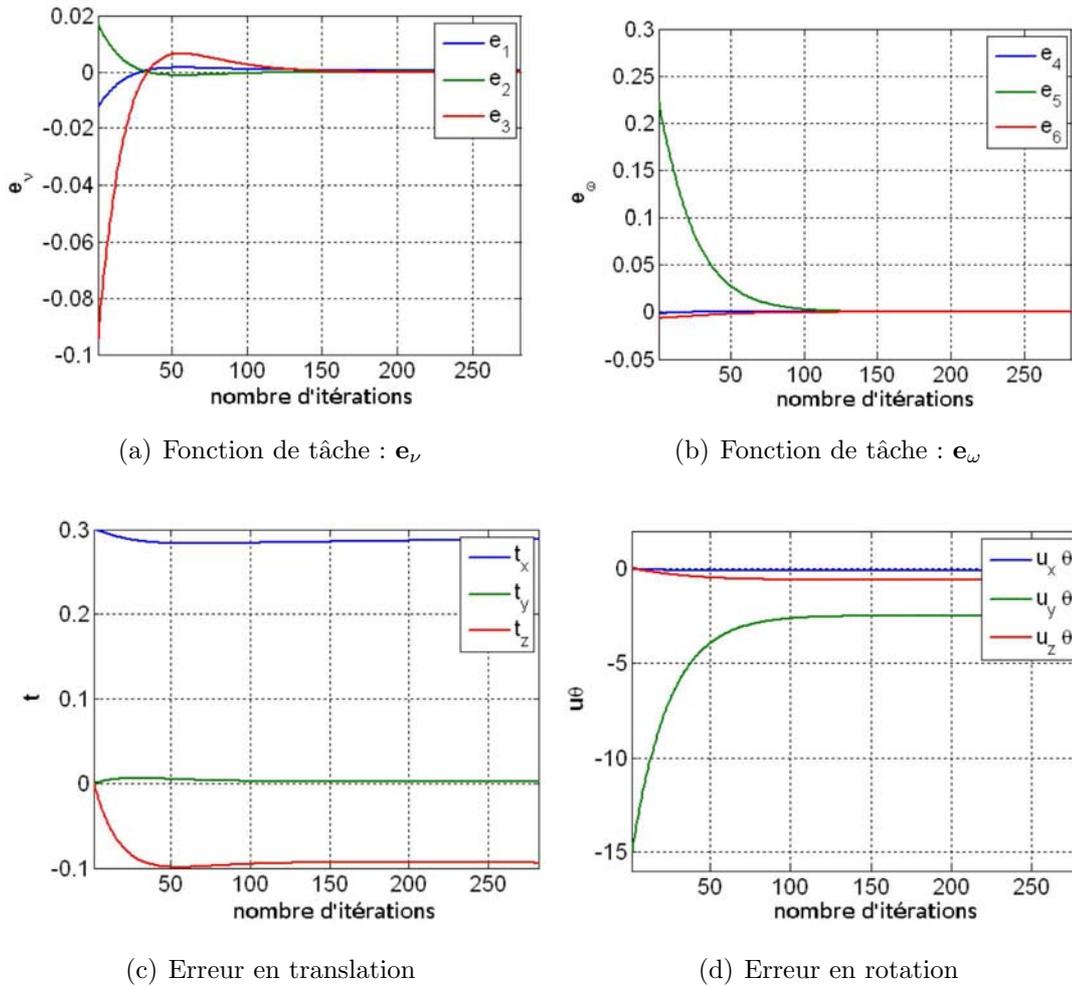


FIG. 2.10 – Simulation du comportement de la fonction de tâche dans l'espace invariant lors d'une erreur sur la profondeur d'un point (valeur réelle divisée par 100) sur 100 points suivis

invariant converge en présence d'une erreur importante de profondeur sur un des points suivis (figures (2.10) (a) et (b)). Mais la position d'arrivée quant à elle est mauvaise, puisque les différences en translation et en rotation entre la position de départ et la position d'arrivée ne sont pas nulles quand la caméra a convergé (figures (2.10) (c) et (d)).

– **Perte des points :**

Les points suivis dans l'image peuvent être perdus suivant la sensibilité de l'algo-

rithme de suivi aux variations d'éclairage, de rotation ou aux changements d'échelle, ou tout simplement si l'objet n'est plus visible dans l'image. La perte de points suivis dans l'image courante, ne perturbe pas la commande tant que ceux-ci ont été bien appariés avec les points de l'image de référence, et qu'ils n'ont pas dérivés au cours du suivi visuel. En revanche, la perte d'un mauvais point, même sur cent points suivis, peut provoquer une discontinuité importante dans la commande, comme l'illustre la figure (2.11) à la dixième itération.

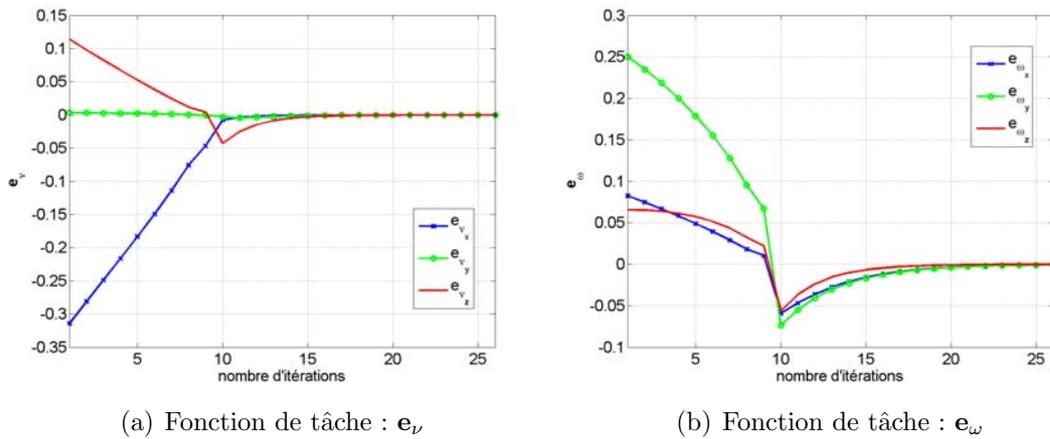


FIG. 2.11 – Simulation du comportement de la fonction de tâche dans l'espace invariant lors de la perte d'un point mal apparié sur 100 suivis

Notons que si la commande dans l'espace invariant ne nécessite pas de calibrer les caméras, elle est en revanche beaucoup plus sensible que la commande 2D au bruit et aux erreurs d'estimations.

2.3 Validation et résultats expérimentaux

2.3.1 Moyens expérimentaux

Pour valider nos travaux nous avons suivi un processus expérimental allant des essais en laboratoire jusqu'aux essais en conditions réelles d'utilisation en mer. La partie expérimentation représente un temps important du travail effectué durant cette thèse, comme en témoigne l'ensemble des moyens expérimentaux utilisés pour nos essais :

- trois bras manipulateur 6ddl (un électrique et deux hydrauliques) ;
- trois paires de caméras CCD (identiques ou différentes) pour des essais dans l'air ;
- une tête stéréo sous-marine 6000m (baptisée IRIS, *Invariant Reconstruction by a Stereovision system*) développée spécialement pour notre application, comportant une caméra fixe, une caméra montée sur pan&tilt et 2 projecteurs à LED ;
- un bassin intérieur et un bassin extérieur d'essais (eau de mer) ;
- le ROV *victor6000* (*Remote Operated Vehicle*) et son instrumentation autorisant des essais en mer jusqu'à 6000m de profondeur (embarqué par le navire océanographique *Pourquoi pas ?* de l'IFREMER) ;
- des dispositifs d'éclairage de forte puissance : projecteur halogène dans l'air, et HMI dans l'eau ;
- de multiples objets de textures, de formes, et de tailles différentes, fabriqués pour expérimenter l'asservissement visuel.

2.3.2 Validation de l'approche

Après avoir validé l'approche de la méthode d'acquisition en simulation, les premières expérimentations ont été menées en laboratoire dans l'air, permettant ainsi de tester en pratique la faisabilité de la trajectoire par asservissement visuel.

2.3.2.1 Conditions expérimentales optimales

Les conditions expérimentales mises en place avaient pour but de limiter les perturbations sur les caméras, et de faciliter le traitement d'image :

- lumière homogène ;
- bonne estimation des paramètres ;
- suivi de plans (objets simples très texturés) ;
- caméras identiques : permet de capturer une image de référence avec des paramètres intrinsèques très proches de la caméras asservie ;
- faible espacement entre les caméras (20cm) ;
- gain faible sur la commande.

2.3.2.2 Méthode générale

Pour les premières expérimentations, le plus important était d'avoir des primitives parfaitement bien appariées, et très bien suivies, afin de tester la répétition de l'asservissement visuel dans les meilleures conditions. En contrepartie, la sélection de points très robustes ne permet pas d'en apparier un grand nombre, et la méthode de suivi employée,

l'ESM [BM04b], ne permet de suivre que des plans dans l'image de façon également très robuste. La commande invariante aux paramètres intrinsèques utilisée, ne pouvant pas être calculée sur des points appartenant au même plan, nous avons utilisé trois plans de normales différentes comme cible 3D. Ainsi la méthode se décompose de la manière suivante :

– **Extraction et appariement de points**

La première étape consiste à sélectionner des zones rectangulaires dans l'image de la caméra gauche. Chaque zone représente une imagerie dans laquelle on va extraire des points SIFT. Puis on extrait d'autres points SIFT dans l'image de la caméra droite, que l'on met ensuite en correspondance avec les points extraits dans les images de la caméra gauche.

– **Suivi de points dans l'image**

L'algorithme ESM employé va suivre le déplacement de chaque petite imagerie sélectionnée dans l'image. Pour chaque déplacement de l'imagerie dû au mouvement de la caméra, l'homographie correspondante est calculée. Il est alors facile de reprojeter les points dans l'image de la caméra asservie.

– **Génération et application de la commande**

La commande invariante présentée précédemment est calculée à partir des coordonnées des points appariés et projetés dans l'espace invariant, de l'estimation des profondeurs et des paramètres intrinsèques.

2.3.2.3 Premiers résultats expérimentaux

Pour ces expérimentations nous avons utilisé le bras manipulateur 6 d.d.l du robot Anis de l'INRIA (Figure (2.12)(a)). Plusieurs déplacements de la tête stéréo ont pu être réalisés, face à une cible composée de trois plans de normales différentes afin d'obtenir une cible 3D (Figure (2.12)(b)) (la commande invariante ne pouvant être calculée à partir d'une cible plane). Dans un premier temps l'estimation de la profondeur des points extraits dans les images a été renseignée manuellement dans le logiciel en les mesurant manuellement, avant de les calculer dans les expérimentations suivantes par triangulation. Ici, chaque ensemble de points présent dans une imagerie se voit attribuer la même profondeur. Les trois plans suivis dans l'image sont représentés par des cadres verts (Figure (2.12)(b)). Des points sont ensuite extraits dans ces plans pour calculer la commande. Notons que la perte du suivi d'un plan entraîne la perte de tous les points qu'il contient.

La figure 2.13 décrit deux asservissements visuels successifs, effectués avec une distance de 20 cm entre les caméras et un angle de 15 degrés. La cible est placée approximativement à 1,3 mètre des caméras. Avec cette géométrie, la tête stéréo décrit une trajectoire circulaire autour de l'objet. La commande est calculée dans un espace invariant aux paramètres intrinsèques des caméras suivant la méthode décrite dans [Mal01a].

Les Figures 2.14 et 2.15 représentent la fonction de tâche qui va définir la commande en vitesse à appliquer à la caméra à un facteur près (équivalent à un gain). À gauche (Figures 2.14(a) et 2.15(a)) sont représentées les composantes de translation, et à droite

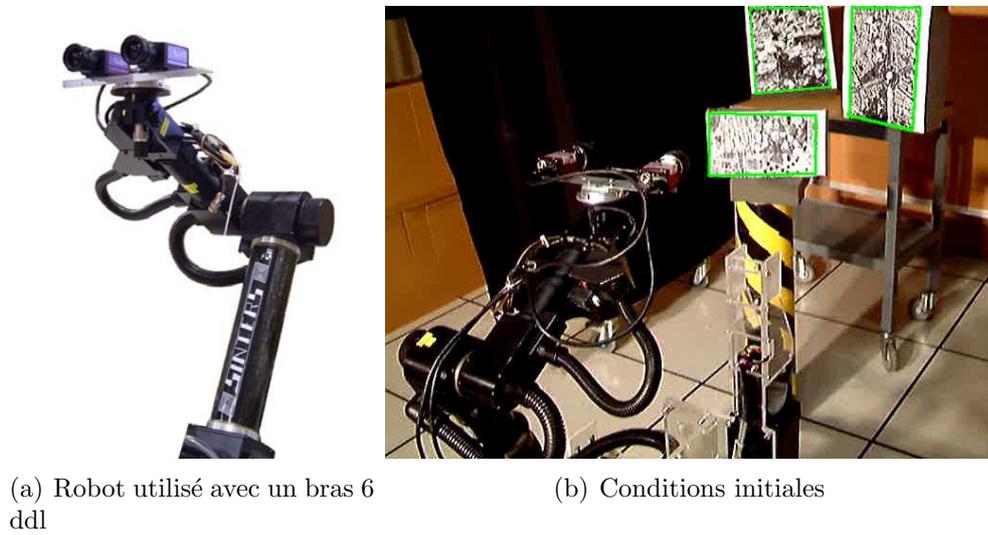


FIG. 2.12 – Conditions expérimentales

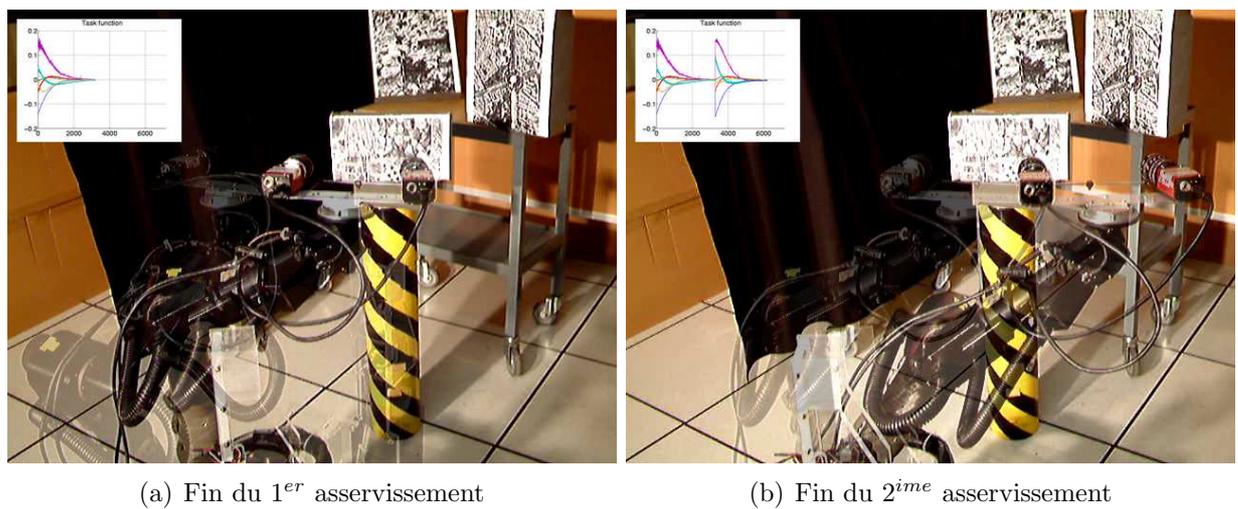


FIG. 2.13 – Asservissement visuel dans l'espace invariant

celles de rotation de la fonction de tâche suivant les axes du repère de la caméra gauche. La commande est ici filtrée par un filtre passe bas pour réduire le bruit provenant des images, et des perturbations du bras. La commande représentée correspond à un asservissement sur 30cm avec un angle de 15° entre les caméras. La commande ensuite calculée représente la vitesse dans l'espace cartésien à appliquer au repère de la caméra pour converger sur la position de référence. Le changement de repère entre le repère caméra et le repère pince est ensuite appliqué pour calculer la commande à appliquer au robot.

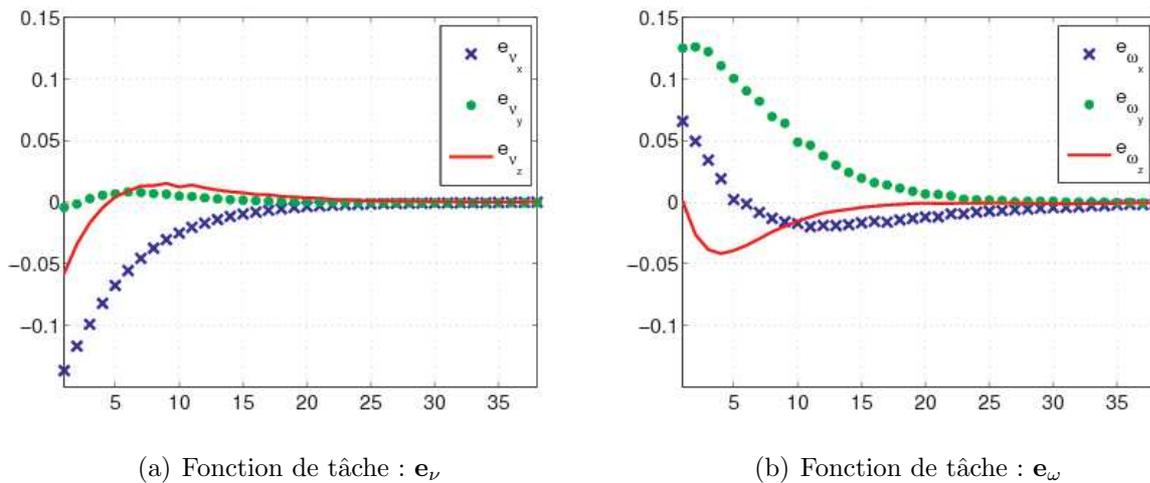


FIG. 2.14 – Fonction de tâche du 1^{er} asservissement

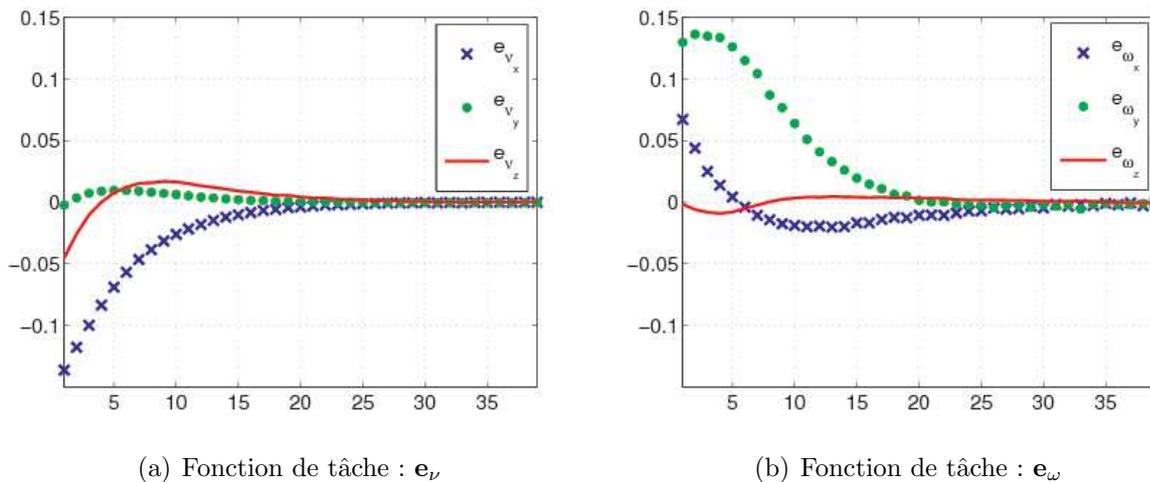
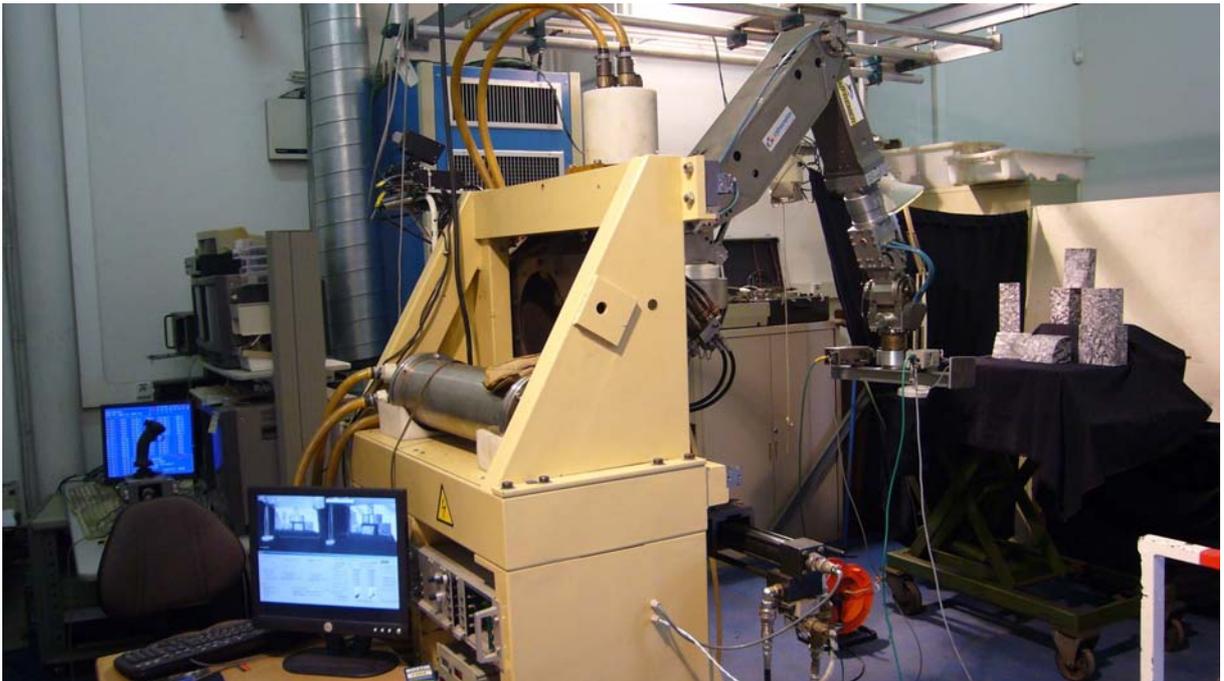


FIG. 2.15 – Fonction de tâche du 2^{ième} asservissement



(a) Conditions expérimentales avec le bras atelier TAO



(b) Bras manipulateur hydraulique 6 d.d.l



(c) Cible composée de morceaux d'écorce de chêne

FIG. 2.16 – Mise en place des expérimentations dans l'air avec le bras atelier TAO de l'Ifremer

2.3.3 Modifications du traitement d'images aux objets 3D

Dans les expérimentations précédentes nous étions dans des conditions optimales pour mener à bien une succession d'asservissements visuels et valider l'approche. Le risque de

perte de points était quasiment nul pour un faible déplacement des caméras, les profondeurs étaient bien estimées, l'appariement robuste, et les déplacements entre les positions des caméras faibles, et le tout sans perturbation d'éclairage ou de visibilité.

Afin de se rapprocher d'avantage des conditions réelles d'utilisation du système final, nous avons utilisé une réplique du bras MAESTRO équipant le robot sous-marin (Figure 2.16(b)). Les cibles planes ont été remplacées par une cible composée d'un ensemble de morceaux d'écorces de chêne liège assemblés sur un cylindre, donnant ainsi un effet naturel de structure rocheuse sous-marine (Figure 2.16(c)). La structure de forme cylindrique se rapproche des cheminées hydrothermales. Nous avons donc changé la méthode de suivi de points, afin de pouvoir suivre indépendamment chaque point sur un objet 3D texturé. La méthode choisie est l'algorithme KLT [TK91] présenté précédemment.

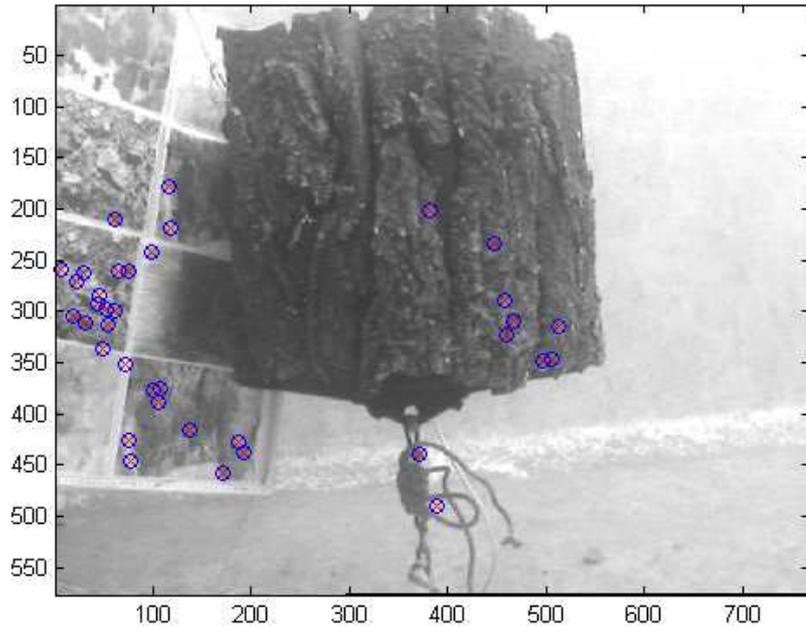
2.3.3.1 Extraction et appariement

Le suivi employé n'étant pas le même, nous avons dû adapter notre méthode au suivi des points KLT, étant donné qu'on ne suit plus des plans mais chaque point indépendamment. Les points SIFT sont toujours utilisés, mais seulement pour permettre d'estimer la matrice fondamentale \mathbf{F} entre les deux images. Les points KLT sont extraits dans une image, puis les points correspondants dans l'autre image sont retrouvés grâce à la contrainte épipolaire, par corrélation sur des images préalablement rectifiées pour faciliter la recherche (la méthode de rectification sera détaillée dans le prochain chapitre). Voici les différentes étapes suivies :

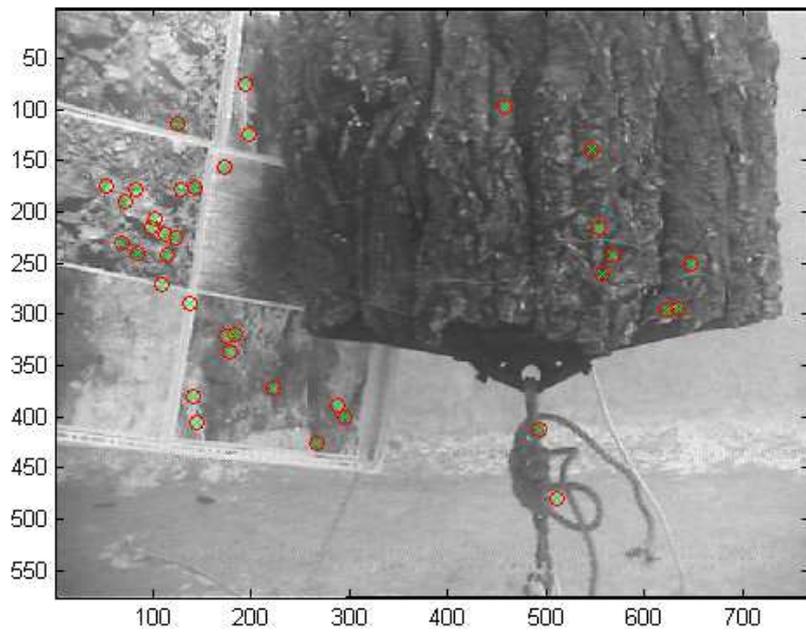
1. Extraction et appariement des points SIFT (Figure 2.17) ;
2. Estimation de la matrice fondamentale \mathbf{F} ;
3. Extraction et appariement des points KLT dans l'image courante (Figure 2.18) ;
4. Rectification des images stéréo à partir de \mathbf{F} et des points SIFT (Figure 2.19, méthode décrite dans le prochain chapitre §3.1.3.1 page 89) ;
5. Recherche par corrélation des points correspondants aux points KLT sur l'image de référence rectifiée, et vérification de la symétrie de la correspondance dans l'image courante ;
6. Transformation des coordonnées des points appariés dans l'image de référence non rectifiée ;
7. Elimination des mauvais appariements (RANSAC).

2.3.3.2 Suivi dans l'image

La méthode de suivi par l'algorithme ESM a été remplacée par l'algorithme KLT pour un suivi individuel des points. Le suivi utilisé est basé sur le travail de Lucas et Kanade, et a été entièrement développé par Tomasi et Kanade. Les points d'intérêt KLT sont très attrayants par leur simplicité. L'extraction et le calcul des déplacements inter-images des points d'intérêt prend 1 ou 2 dixièmes de secondes sur un PC doté d'un processeur 3GHz. Les points d'intérêts sont localisés en examinant la plus petite valeur propre de la matrice d'autocorrélation (2×2) en chaque point de l'image, telle qu'elle soit supérieure à un seuil

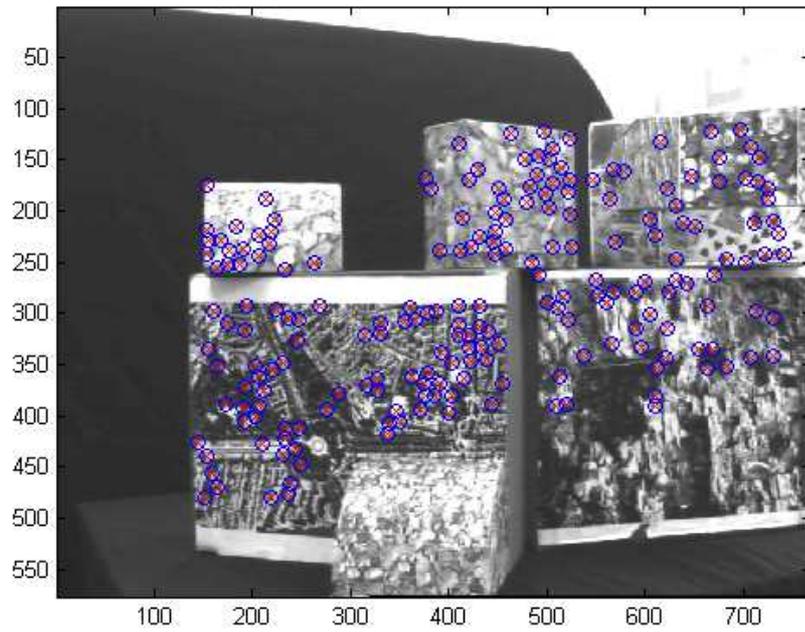


(a) Image caméra gauche

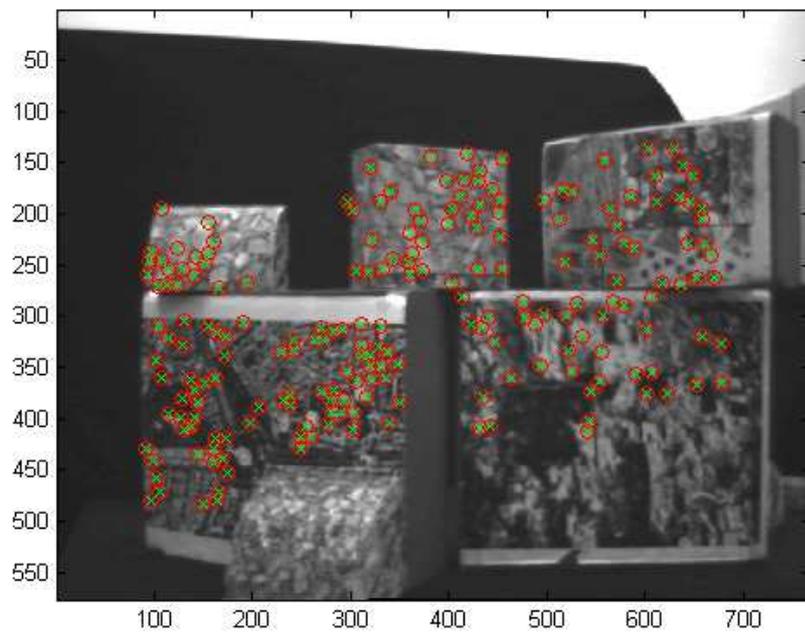


(b) Image caméra droite

FIG. 2.17 – Points SIFT mis en correspondance sur des images sous-marines

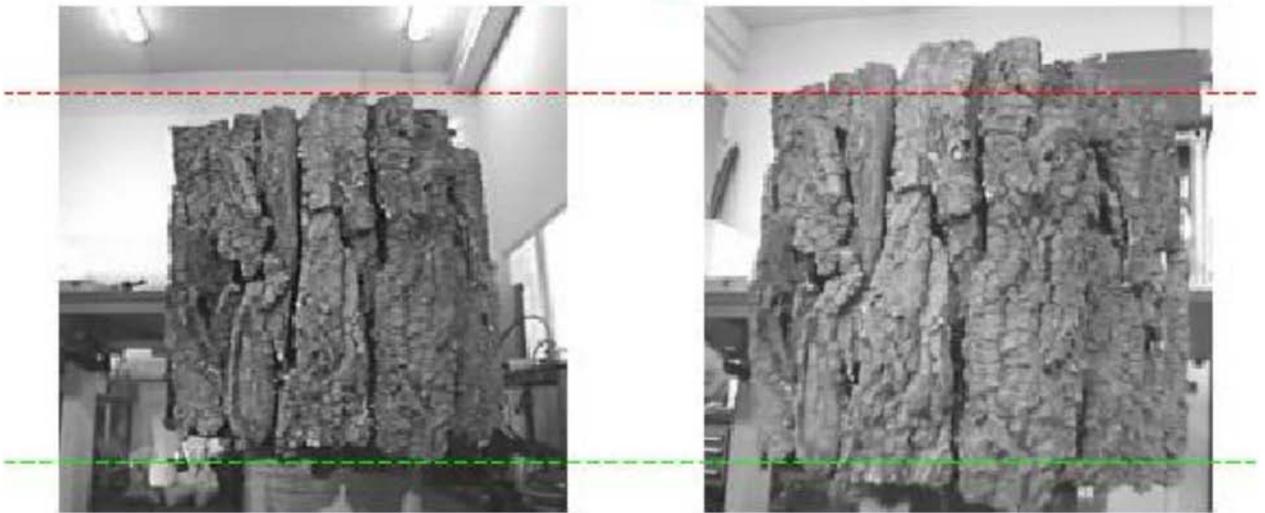


(a) Image caméra gauche



(b) Image caméra droite

FIG. 2.18 – Points KLT appariés dans les images stéréo



(a) Images caméras gauche et droite



(b) Images après rectification

FIG. 2.19 – Rectification des images issues de la tête stéréo

fixé. Pour une valeur du seuil égale à 1, le nombre de points d'intérêt dans une image de taille 352×288 (Common Intermediate Format, CIF) est d'environ 500. Les points sont ensuite suivis en utilisant la méthode de minimisation itérative de type Newton-Raphson. Les points caractéristiques KLT sont répartis de manière non-homogène dans l'image. Ils sont généralement concentrés dans les zones texturées. Il faut en choisir suffisamment pour avoir un recouvrement correct de tous les objets dans l'image. Chaque point perdu durant le suivi n'est pas remplacé par un nouveau point extrait dans l'image. La méthode de suivi utilise un mode séquentiel, qui consiste à ne suivre les points qu'entre deux images successives. A chaque itération, la seconde image est enregistrée une fois le suivi effectué, puis est utilisée à la prochaine itération comme première image. Les erreurs de suivi d'une image sur l'autre pouvant s'accumuler, la vérification de la similarité par une simple mise en correspondance avec translation de la fenêtre n'est pas suffisante. Ainsi pour détecter

des primitives qui ont dérivé au cours du suivi, on vérifie la similarité entre la fenêtre autour de la primitive dans la première image et celle autour de la primitive dans l'image courante avec une transformation affine, puisque l'on aura une déformation perspective du contenu de l'image après un déplacement de la caméra (Figure 2.20).

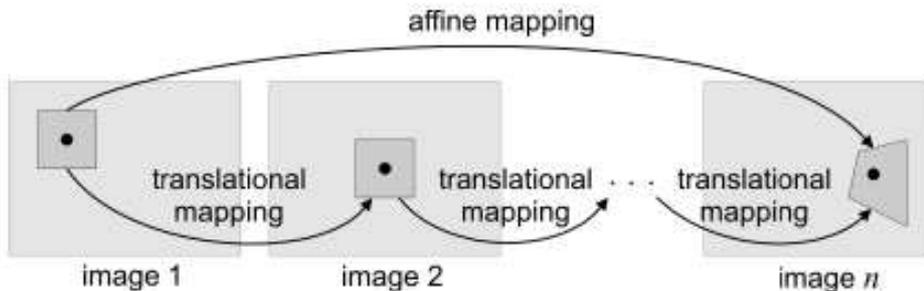


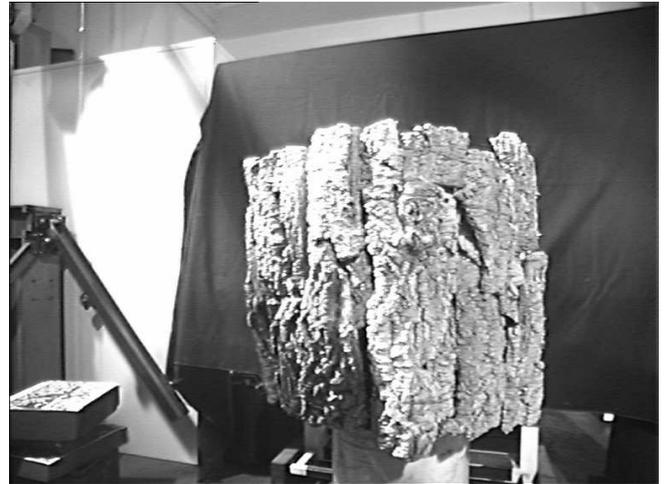
FIG. 2.20 – Vérification de la similarité [Bir07]

Les figures 2.21, 2.22, et 2.23 montrent le résultat d'un asservissement visuel avec la commande invariante et le suivi KLT, en utilisant des caméras identiques. L'image d'arrivée est nettement décalée par rapport à l'image de référence comme le montre la figure 2.21.d, bien que l'on soit quand même au voisinage de la position de référence. L'explication de ce décalage, que l'on retrouve au niveau des primitives de référence et d'arrivée sur la figure 2.22.a, alors même que l'on a convergé est illustrée par la figure 2.22.b. En effet, on remarque nettement dans l'espace invariant que certains points sont éloignés de leur position de référence, ce qui indique qu'ils ont dérivé. On note également la sensibilité de la commande invariante par rapport à une commande 2D classique, en comparant la distance entre les points courants et les points de référence dans l'espace image (figure 2.22.a) par rapport à l'espace invariant (figure 2.22.b). La commande (figure 2.23) quant à elle a bien convergé vers 0, que ce soit en translation ou en rotation. Cependant la commande étant extrêmement faible en fin d'asservissement, et n'ayant pas mis de gain adaptatif, on peut supposer qu'à la dérive des points s'ajoute la difficulté pour le bras manipulateur d'intégrer de très faibles déplacements.

Le choix des méthodes d'extraction, d'appariement, et de suivi visuel, est déterminant pour le contrôle du robot, puisque les points suivis et appariés dans les images vont servir à calculer la commande. L'influence des erreurs de traitement d'images sur la commande sera détaillée dans la prochaine partie.



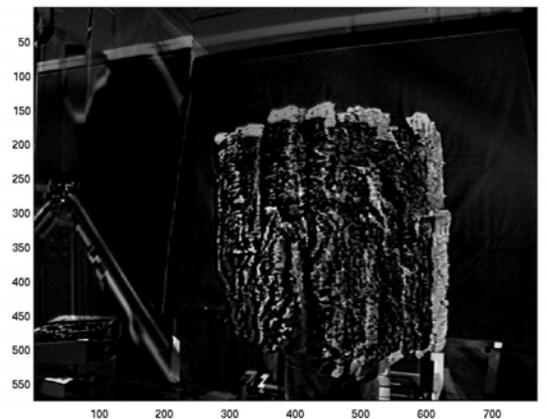
(a) Image de la caméra gauche avant l'asservissement visuel



(b) Image de référence de la caméra droite

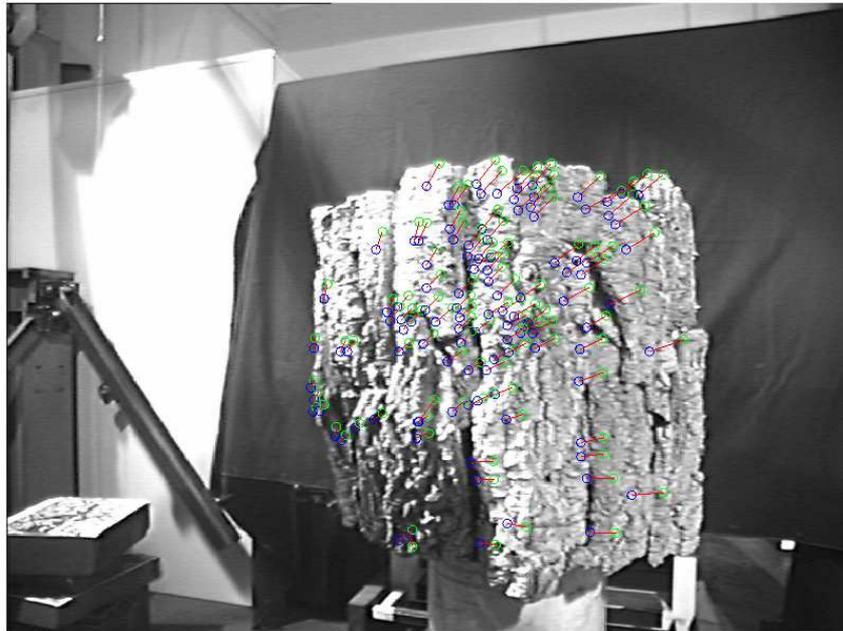


(c) Image de la caméra gauche après avoir convergé

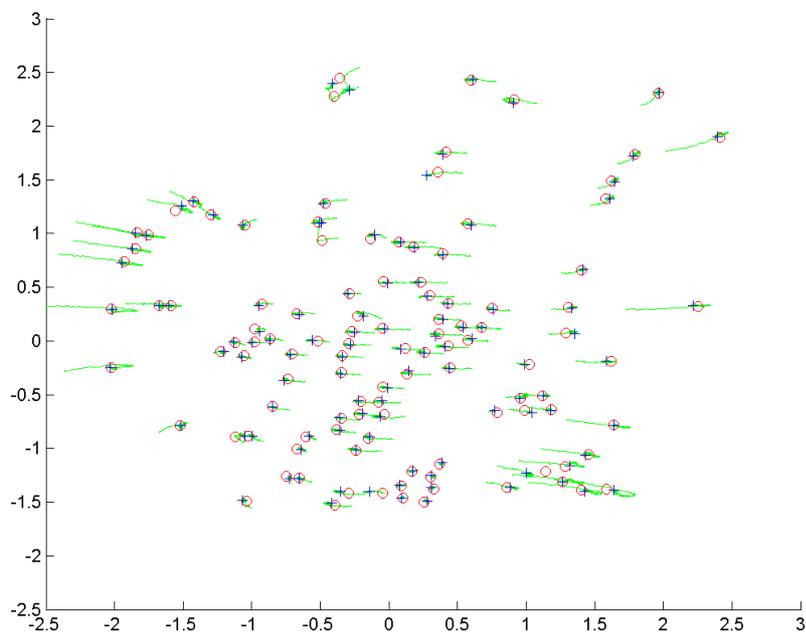


(d) Différence entre l'image de référence (b) et l'image courante à la fin de l'asservissement (c)

FIG. 2.21 – Résultats de l'asservissement visuel entre deux caméras identiques en utilisant la commande invariante. La géométrie des caméras est de 30cm avec un angle de 15° .



(a) Image de référence avec en vert les points de référence, et en bleu les points suivis à la fin de l'asservissement



(b) Projection des points dans l'espace invariant. En rouge sont représentés les points de référence, en bleu les points suivis après avoir convergé, et en vert la trajectoire des points suivis.

FIG. 2.22 – Représentation des points de référence et suivis dans l'image de référence et dans l'espace invariant, après avoir convergé par asservissement visuel.

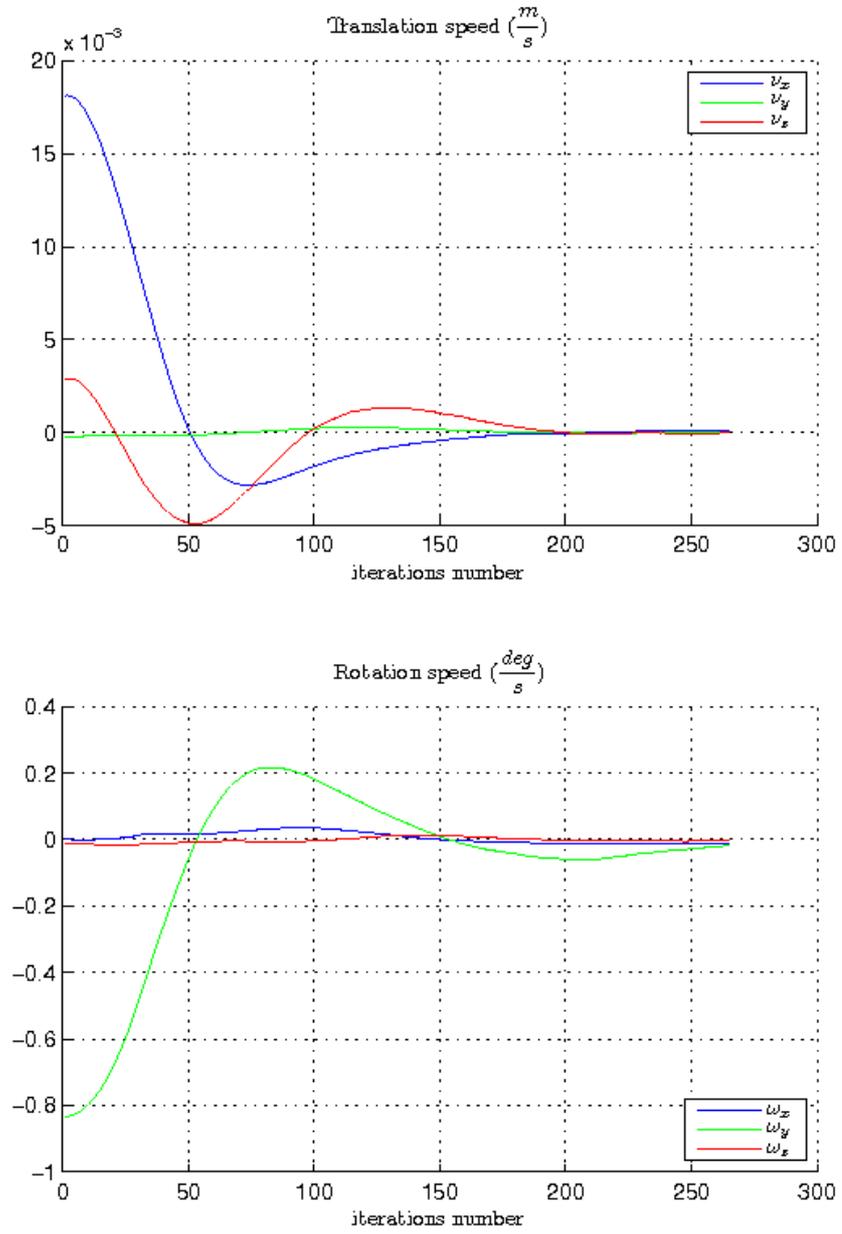


FIG. 2.23 – Commande invariante filtrée appliquée à la caméra gauche

2.3.4 Contrôle du déplacement du robot

Le contrôle du déplacement du robot peut se faire soit par asservissement visuel, soit par programmation de la position articulaire d'arrivée de chaque axe du bras (Programmes Robots). Dans chacun de ces cas la précision et la répétabilité sont différentes et varient en fonction des erreurs de réglages du bras ou de traitement d'images.

2.3.4.1 Impact de l'estimation des paramètres intrinsèques, et des profondeurs des points sur l'asservissement visuel

La commande invariante aux paramètres intrinsèques converge même avec 50% d'erreur sur \mathbf{K} , pour un déplacement de 30cm et une rotation de 10° (Figure 2.24). Le transitoire au niveau de la commande peut osciller d'avantage, ce qui se traduit par la répétition d'un mouvement d'avant en arrière sur les caméras, tout en convergeant vers la bonne position.

Par contre, la commande invariante est sensible à l'estimation des profondeurs des points tout comme la commande 2D. Celle-ci a beaucoup de mal converger au-delà de 20% d'erreur sur les profondeurs. Il est essentiel de conserver le rapport des distances en profondeur, mais une simple estimation suffit pour converger.

2.3.4.2 Influence de mauvais appariements et de la dérive de points sur le comportement de l'asservissement visuel

Le comportement de la commande observé en cas de mauvais appariements ou de dérives des points est conforme à celui observé en simulation, la caméra ne converge pas sur la bonne position. Le risque de dérive des points en expérimentation est important avec la méthode de suivi KLT, puisque l'algorithme met à jour l'image de référence avec l'image courante après chaque itération. Cependant en faisant un test de corrélation autour des points avec la première image du suivi, on peut réussir à éliminer quelques mauvais points, et ainsi corriger la commande.

2.3.4.3 Précision et répétabilité de l'asservissement visuel (commande 2D/invariante)

Afin de tester la fiabilité des différentes méthodes de positionnement des caméras, il est important d'étudier leur répétabilité. Le problème dans notre cas est d'établir une vérité de terrain. Pour répéter un asservissement visuel, nous sommes obligés de repositionner le bras dans la même configuration de départ par un programme robot (asservissement en position du bras manipulateur). Or, l'asservissement du bras par rapport à une consigne articulaire n'est pas parfait. Toutes les valeurs relevées pour les déplacements par asservissement visuel contiennent donc une erreur due au repositionnement du bras dans la configuration de départ. Les résultats d'asservissement visuel doivent donc être interprétés en tenant compte de la précision des programmes robots.

Les valeurs relevées pour étudier la répétabilité du positionnement des caméras, représentent les positions articulaires des 6 axes du bras manipulateur TAO, ainsi que la position de l'organe effecteur du bras dans l'espace cartésien seulement dans le cas des

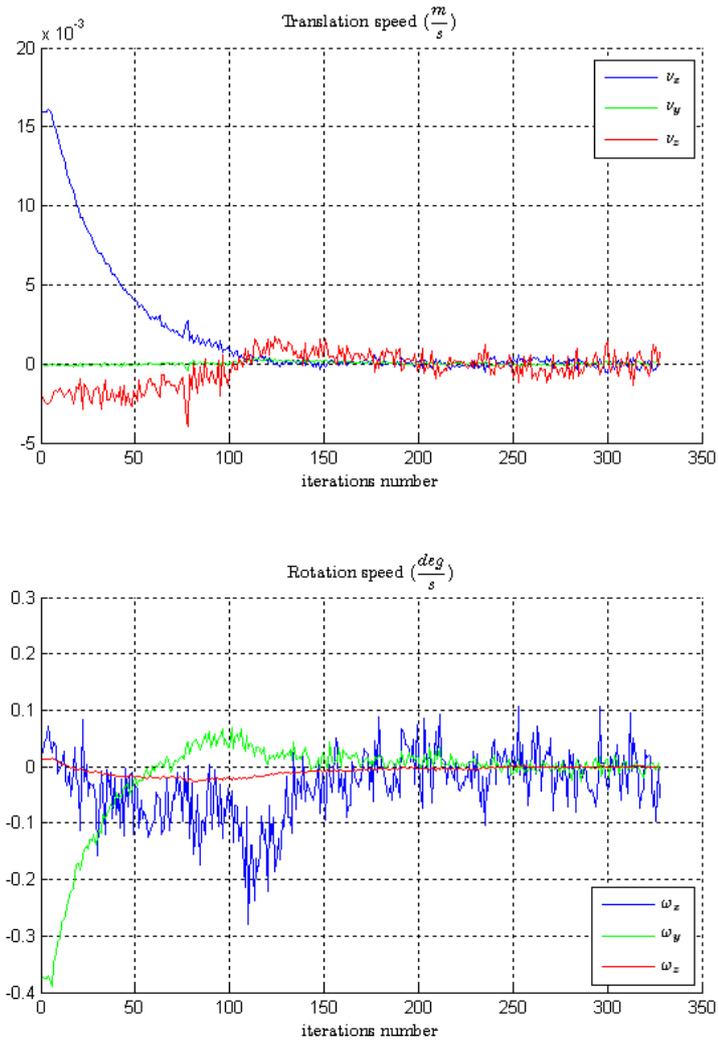


FIG. 2.24 – Commande invariante avec 50% d'erreur sur \mathbf{K}

asservissements visuels (obtenue en appliquant le modèle géométrique du bras). Ces valeurs articulaires sont relevées directement sur un écran d'ordinateur à partir de la lecture des résolveurs du bras, qui transmettent le changement d'angle d'un axe en une valeur électrique. Pour tous les résultats expérimentaux présentés dans ce paragraphe, la géométrie de la tête stéréo est de 30 cm entre les caméras, avec un angle de 10° de la caméra droite par rapport à la caméra gauche.

– **Programmes robots**

La même trajectoire décrite par asservissement visuel peut être générée par “programmes robot”, lesquels consistent à programmer les positions des six axes de rotation du bras manipulateur. Ainsi, le robot accomplit un asservissement électrohydraulique de position grâce aux informations provenant de ses capteurs, les résolveurs. Ainsi cette méthode ne requiert pas d'informations visuelles. Le plus gros inconvénient est la programmation de tous les axes du bras pour chaque position de prise de vue, ainsi que pour chaque modification de la géométrie de la tête stéréo. Cependant, cette méthode peut tout de même être utilisée pour placer le bras dans une position initiale avant de décrire la trajectoire par asservissement visuel, ceci permettant d'avoir une répétabilité lors des expérimentations, et d'assurer un placement optimal des caméras par rapport à l'espace de travail du bras disponible pour exécuter la trajectoire. Néanmoins, l'avantage de ces trajectoires pré-planifiées repose sur la robustesse de la méthode, puisque la précision de la trajectoire dépend uniquement de la précision et du réglage du bras manipulateur ainsi que de la mesure de la géométrie de la tête stéréo, et non pas de l'information contenue dans les images des caméras.

Le tableau 2.1, présente les résultats de cinq repositionnements du bras du robot par rapport à une même consigne donnée par programme robot. Les valeurs relevées sur chacun des six axes sont données par la lecture des résolveurs du bras. A la suite des résultats expérimentaux ont été calculés la moyenne des erreurs absolues et relatives, ainsi que l'écart type. On note une erreur moyenne absolue d'environ 2° sur l'axe 4 par rapport à la consigne, mais avec un écart type presque 10 fois plus faible. Ceci montre que le bras a une bonne répétabilité, mais avec des erreurs de calibration sur certains axes, puisqu'on retrouve la même observation sur les axes 1, 3 et 6.

axe	consigne (deg)	1	2	3	4	5	δ_{abs}	δ_{rel}	σ_{rob}
θ_1	34,77	35,69	35,58	35,60	35,59	35,63	0,848	$2,44 \times 10^{-2}$	$3,97 \times 10^{-2}$
θ_2	36,36	36,35	36,36	36,35	36,36	36,38	0,006	$2,2 \times 10^{-4}$	$1,10 \times 10^{-2}$
θ_3	-5,46	-5,32	-5,31	-5,35	-5,34	-5,32	1,132	$2,42 \times 10^{-2}$	$1,47 \times 10^{-2}$
θ_4	0,11	2,28	2,16	1,62	2,08	2,10	1,95	17,618	$2,25 \times 10^{-1}$
θ_5	-34,10	-33,99	-33,97	-34,22	-33,56	-34,05	0,19	$5,57 \times 10^{-3}$	$2,18 \times 10^{-1}$
θ_6	-49,78	-50,03	-50,05	-50,30	-50,02	-50,04	0,308	$6,19 \times 10^{-3}$	$1,06 \times 10^{-1}$

TAB. 2.1 – Répétabilité et précision du positionnement du bras par programmes robots

– **Commande 2D**

Dans le cas des asservissements visuels, il est difficile d’avoir une consigne articulaire servant de référence, étant donné que la commande est calculée dans l’image. Les valeurs relevées sont les positions articulaires (tableau 2.2), et la position et l’orientation du repère pince du bras (tableau 2.3). L’élément de comparaison dans le tableau 2.2 avec les programmes robots est l’écart type, qui permet de voir que l’asservissement est un peu moins bon dans la répétabilité du positionnement du bras.

axe	1	2	3	σ_{2D}
θ_1	20,73	20,92	21,08	$1,43 \times 10^{-1}$
θ_2	43,32	43,72	44,74	$5,98 \times 10^{-1}$
θ_3	-23,81	-24,15	-24,64	$3,41 \times 10^{-1}$
θ_4	4,47	4,75	3,76	$4,17 \times 10^{-1}$
θ_5	-23,44	-24,21	-24,20	$3,61 \times 10^{-1}$
θ_6	-29,13	-29,09	-28,06	$4,95 \times 10^{-1}$

TAB. 2.2 – Mesures de la position des axes du bras après un asservissement visuel par commande 2D

	1	2	3	σ_{2D}
$t_x(\text{m})$	0,834	0,829	0,824	$4,08 \times 10^{-3}$
$t_y(\text{m})$	-0,308	-0,309	-0,311	$1,25 \times 10^{-3}$
$t_z(\text{m})$	0,509	0,505	0,493	$6,80 \times 10^{-3}$
Gisement (deg)	-134,64	-134,57	-140,38	2,72
Pendage (deg)	4,24	4,64	4,97	$2,98 \times 10^{-1}$
R_z(deg)	-49,62	-49,20	-43,07	2,99

TAB. 2.3 – Mesures des coordonnées cartésiennes du repère pince après un asservissement visuel par commande 2D

– **Commande invariante**

La base invariante servant à faire une projection des points extraits dans l’image dans un espace invariant aux paramètres intrinsèques [Mal04b], est calculée à partir de tous les points et à chaque itération du suivi. Ainsi, l’erreur sur les coordonnées d’un point qui a dérivé dans l’image va se répercuter sur le calcul de la base et donc sur tous les points projetés dans l’espace invariant. Ceci rend beaucoup plus sensible la commande invariante à la dérive des points que la commande 2D. Ainsi, la solution adoptée a été de ne calculer la base invariante qu’à partir des points initiaux, en supposant qu’il n’y ait pas de mauvais appariements de points. L’erreur reste donc uniquement sur le point qui a dérivé, ce qui limite sa répercussion sur la commande.

Au final on remarque donc dans les expérimentations avec les tableaux 2.3 et 2.5, des performances très proches au niveau du positionnement du repère pince dans

l'espace cartésien, entre la commande 2D et la commande invariante.

axe	1	2	3	σ_{inv}
θ_1	21,67	21,43	20,67	$4,26 \times 10^{-1}$
θ_2	44,06	44,41	45,06	$4,14 \times 10^{-1}$
θ_3	-24,40	-25,02	-25,43	$4,23 \times 10^{-1}$
θ_4	4,20	5,27	5,69	$6,27 \times 10^{-1}$
θ_5	-24,38	-24,04	-24,85	$3,32 \times 10^{-1}$
θ_6	-29,38	-31,98	-30,68	1,06

TAB. 2.4 – Mesures de la position des axes du bras après un asservissement visuel par commande invariante

	1	2	3	σ_{inv}
$\mathbf{t}_x(\mathbf{m})$	0,821	0,817	0,820	$1,70 \times 10^{-3}$
$\mathbf{t}_y(\mathbf{m})$	-0,319	-0,313	-0,297	$9,29 \times 10^{-3}$
$\mathbf{t}_z(\mathbf{m})$	0,501	0,498	0,491	$4,19 \times 10^{-3}$
Gisement (deg)	-138,40	-133,18	-132,90	2,53
Pendage (deg)	5,07	5,06	5,23	$7,79 \times 10^{-3}$
$\mathbf{R}_z(\text{deg})$	-45,47	-52,37	-52,00	3,17

TAB. 2.5 – Mesures des coordonnées cartésiennes du repère pince après un asservissement visuel par commande invariante

En conclusion, on peut donc dire que les deux commandes utilisées pour l'asservissement visuel donnent sensiblement les mêmes résultats (tableau 2.6). La commande 2D est sensible aux erreurs de calibration, tandis que la commande invariante est plus sensible au bruit. Par contre le repositionnement par programme robot donne de meilleurs résultats que l'asservissement visuel. Néanmoins, l'avantage de l'asservissement visuel est de pouvoir s'affranchir des erreurs de calibration et de la cinématique du bras, puisque la commande est générée dans l'image. La position de référence est entièrement définie par la géométrie de la tête stéréo. Les programmes robots doivent être modifiés chaque fois que l'on modifie la trajectoire, et en fonction du bras manipulateur utilisé. Au final l'asservissement visuel à l'aide du système stéréo représente une solution beaucoup plus flexible pour l'étude d'objets naturels divers, mais avec une moins bonne précision que les programmes robots.

2.3.5 Génération de trajectoires

La figure 2.25 représente la comparaison des positions des caméras obtenues par asservissement visuel en utilisant la commande 2D, avec la trajectoire théorique. Les caméras vertes représentent les différentes positions de la caméra droite de la tête stéréo, et donc également les positions de référence à atteindre pour la caméra gauche représentée en rouge. On remarque que les positions de la caméra gauche ne coïncident pas avec les

axe	σ_{rob}	σ_{2D}	σ_{inv}
θ_1	$3,97 \times 10^{-2}$	$3,60 \times \sigma_{rob}$	$10,74 \times \sigma_{rob}$
θ_2	$1,10 \times 10^{-2}$	$54,58 \times \sigma_{rob}$	$37,82 \times \sigma_{rob}$
θ_3	$1,47 \times 10^{-2}$	$23,18 \times \sigma_{rob}$	$28,81 \times \sigma_{rob}$
θ_4	$2,25 \times 10^{-1}$	$1,85 \times \sigma_{rob}$	$2,79 \times \sigma_{rob}$
θ_5	$2,18 \times 10^{-1}$	$1,66 \times \sigma_{rob}$	$1,53 \times \sigma_{rob}$
θ_6	$1,06 \times 10^{-1}$	$4,65 \times \sigma_{rob}$	$9,97 \times \sigma_{rob}$

TAB. 2.6 – Comparaison des écarts types sur le positionnement du bras par asservissement visuel par rapport aux écarts types sur le positionnement par programmes robots

positions de la caméra droite. Il y a donc une dérive des caméras qui s'éloignent des positions théoriques, due aux problèmes vus précédemment de bruit dans l'image, de mauvais appariements, et de dérive des points.

2.3.6 Développement d'un prototype opérationnel

Conception et réalisation du système

La tête stéréo sous-marine IRIS utilisée pour les expérimentations est composée de deux caméras et de deux projecteurs montés sur un bâti permettant d'ajuster l'ensemble. Le système est dans une structure en aluminium rendant possible la saisie par la pince du bras suivant deux positions (dessus ou derrière), et protégeant l'ensemble contre les chocs (figure 2.26). La tête stéréo est utilisable jusqu'à une profondeur de 6000 mètres, et est suffisamment compacte pour être embarquée dans le panier du ROV avec d'autres instruments.

Une caméra est fixe, tandis que l'autre est contrôlable par liaison série RS232 à partir d'un mécanisme "pan and tilt". Ainsi les caméras peuvent être positionnées de façon à reproduire les trajectoires faites en simulation et en expérimentation dans l'air. Les deux caméras sont placées en caisson étanche permettant ainsi de résister à la pression jusqu'à 6000 mètres de profondeur.

Les projecteurs sont à LED, et viennent en complément des projecteurs HMI placés sur le ROV. Chacun fournit 35 W (5 W par LED), pour un volume réduit de 10×10 cm. Ils sont placés à côté des caméras afin de réduire l'ombre portée du bras sur la scène due aux projecteurs du ROV.

Enfin une boîte de jonction permet de réunir tous les câbles provenant des caméras et des projecteurs en un seul, qui fait le lien avec le ROV. Un système d'élastiques permet de sortir et de rentrer la tête stéréo dans le panier du ROV avec le bras manipulateur, sans être gêné par le câble, et sans emmêler celui-ci.

Essais préliminaires dans l'eau

Avant d'utiliser le système sous-marin en mer, des essais préliminaires ont été effectués en bassin, puis en piscine d'eau de mer de jour et de nuit (figure 2.27). Ces essais prélimi-

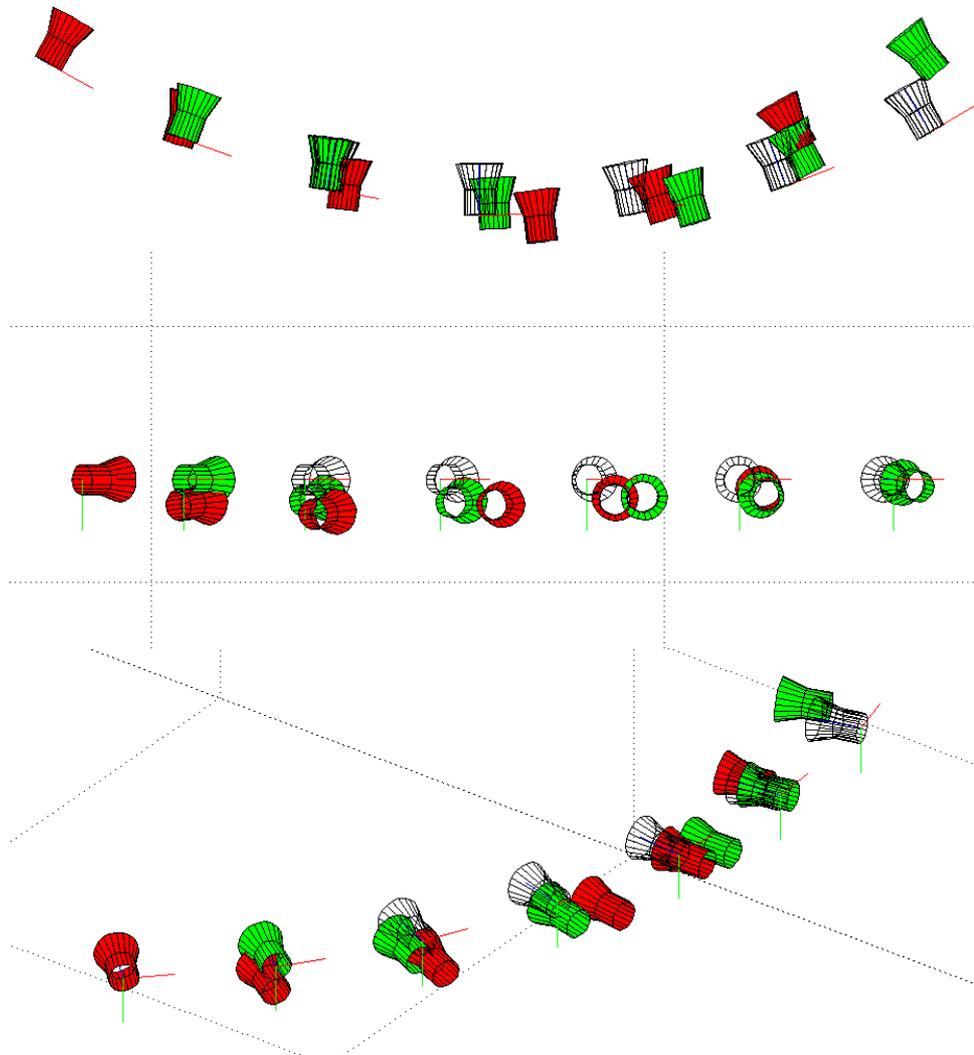


FIG. 2.25 – Trajectoire par asservissement visuel avec la commande 2D appliquée au bras TAO. Les caméras transparentes représentent les positions théoriques, celles en vert correspondent à la caméra droite de la tête stéréo, et celles en rouge aux positions de la caméra gauche. La géométrie de la tête stéréo est de 30 cm entre les caméras, et de 10° sur la caméra droite.

naires ont permis de déceler un problème d'étanchéité d'un câble, de faire la calibration des caméras dans l'eau, de tester la portée des projecteurs par rapport aux projecteurs HMI, et de faire l'acquisition d'images sous-marines.

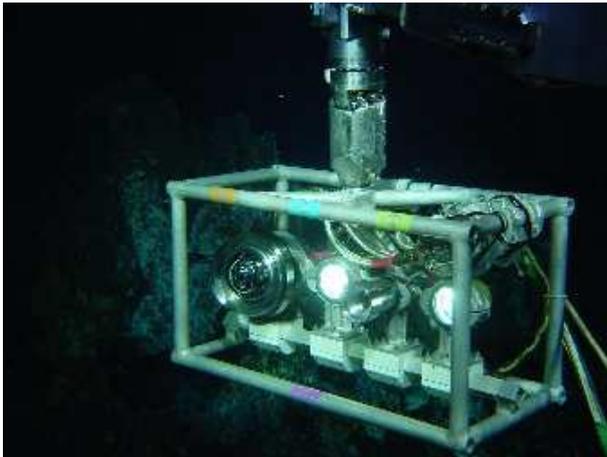
Des essais en mer ont ensuite été effectués avec le ROV *victor6000*, embarqué sur le navire océanographique *Pourquoi Pas ?*, durant la campagne en mer Momareto en août 2006 au



FIG. 2.26 – Tête stéréo IRIS composée d’une caméra mobile, d’une caméra fixe, de deux projecteurs à LED, et d’une boîte de jonction (absente sur la photo en haut à droite). La structure en aluminium fait 710 mm de largeur, 479 mm de hauteur, et 350 mm de profondeur.

sud des Açores, dans le cadre du projet européen EXOCET/D. Le ROV est alimenté depuis le bateau par un ombilical de 8500 m avec une puissance de 1500 W en 24 VCC, qui à son tour alimente la tête stéréo en 24 V (figure 2.28). Les images vidéos des caméras sont remontées au bateau par l’ombilical en temps réel. Le “pan and tilt” est directement commandable depuis la surface.

Le site d’expérimentation en mer est la station Lucky Strike, qui est un des sites hydrothermaux les plus étendus de ceux qui ont été visités à ce jour dans l’océan mondial (figure 2.29.a). Sa profondeur moyenne est de 1700 mètres. Plusieurs séries d’images ont été acquises par les caméras par programmes robots (figure 2.29.b), afin de les exploiter par la méthode de reconstruction 3D détaillée dans la prochaine partie. L’asservissement visuel n’a pas pu être utilisé pour faire l’acquisition d’images durant ces expérimentations, en raison d’un problème de mise au point sur l’une des deux caméras rendant l’image floue et donc inexploitable. La trajectoire effectuée pour l’acquisition des images est représentée sur la figure 2.30. L’espace de travail du bras MAESTRO permet d’avoir 21 positions différentes des caméras pour une position du ROV posé sur le fond, avec une géométrie de la tête stéréo de 30 cm entre les caméras et de 15° sur la caméra droite.



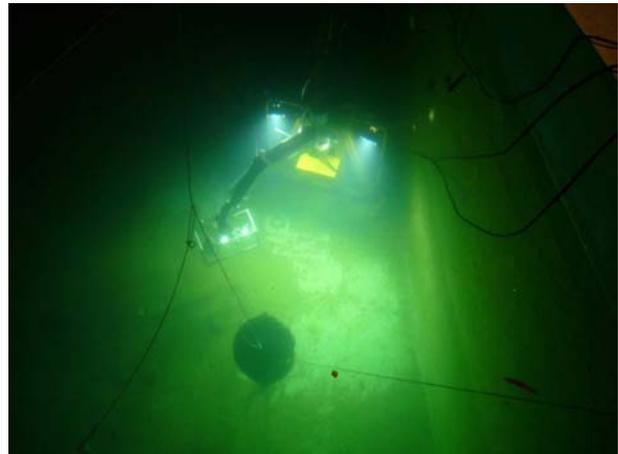
(a) Tête stéréo IRIS



(b) Essai de la tête en bassin intérieur

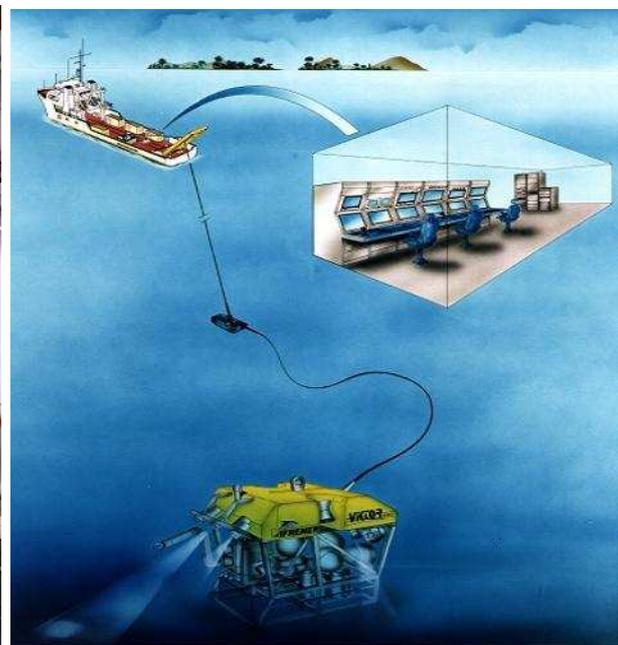


(c) Expérimentation avec le bras manipulateur et IRIS en bassin extérieur



(d) Expérimentation de nuit en bassin extérieur avec 2 projecteurs HMI

FIG. 2.27 – Mise en place des expérimentations dans l'eau avec le bras atelier TAO

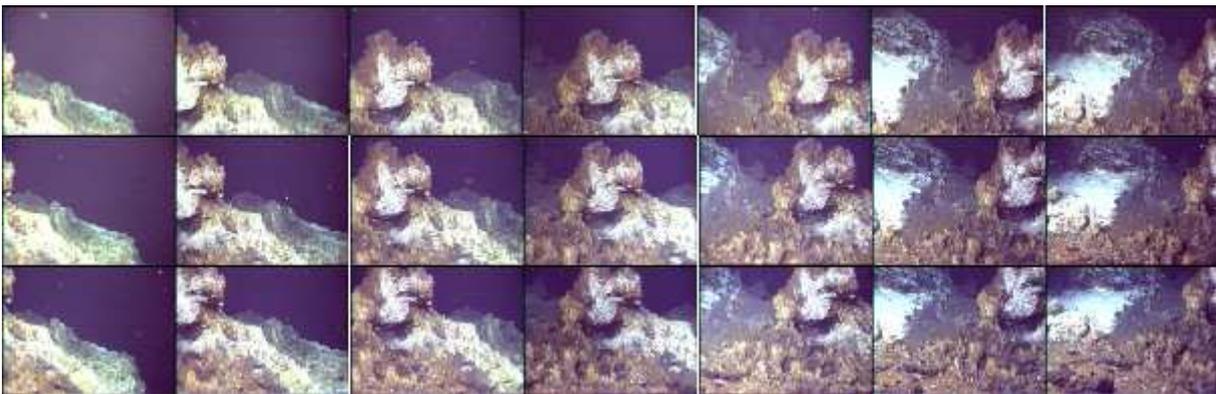


(a) ROV *victor6000* avec la tête IRIS te- (b) Communication entre le ROV et le PC en surface
nue par le bras MAESTRO

FIG. 2.28 – Expérimentations avec le ROV *victor6000*



(a) Zone d'observation sur la station Lucky strike aux Açores



(b) Mosaique d'images prises suivant la trajectoire effectuée par programmes robots

FIG. 2.29 – Acquisition des images servant à la reconstruction sur un site à 1700m de profondeur

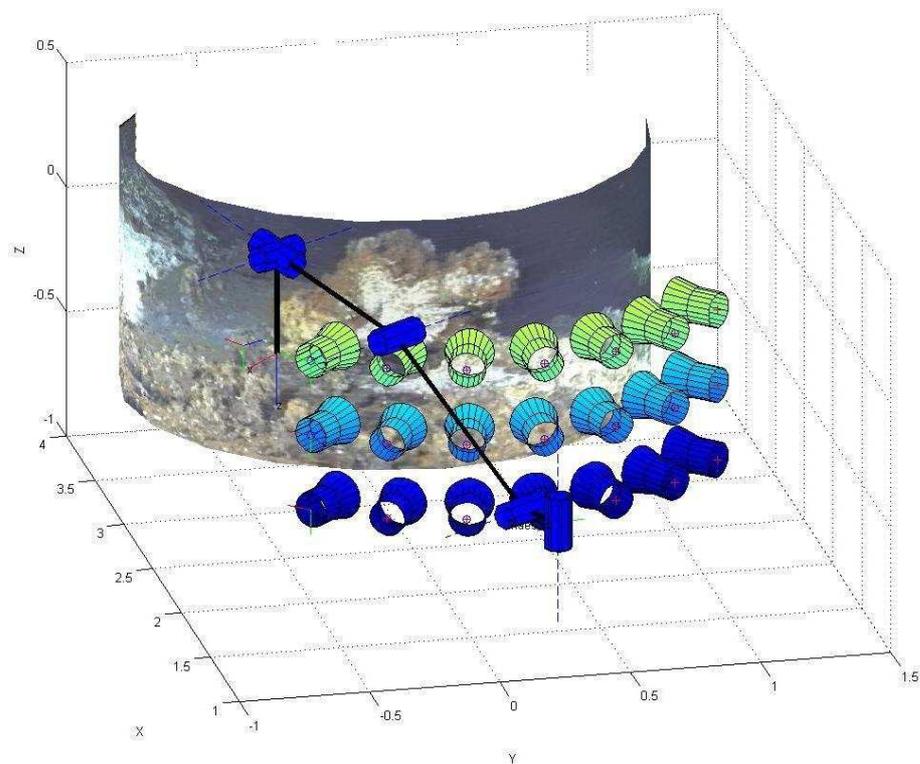


FIG. 2.30 – Trajectoire théorique pour les essais en mer de la tête stéréo IRIS effectués avec le bras MAESTRO

2.4 Conclusion

L'objectif principal de cette thèse est de développer une méthode de reconstruction 3D d'objets sous-marins qui puisse être utilisée en conditions opérationnelles. Les expérimentations ont donc été une étape essentielle pour passer de la théorie avec les simulations à la pratique en conditions réelles.

Les simulations ont permis à la fois de valider l'approche (trajectoires par asservissement visuel), de mettre en place les expérimentations (dimensionnement de la tête stéréo en fonction de l'objet à reconstruire, programmation des positions du bras sur la trajectoire), et d'analyser les dysfonctionnements dans les expérimentations (dérive ou perte des points, bruit dans l'image, mauvais appariements).

Les essais dans l'air ont été les plus nombreux, ceux dans l'eau nécessitant beaucoup de logistique. Les principaux problèmes mis en avant dans les expérimentations sont liés à l'appariement et au suivi des primitives, le calcul de la commande restant ensuite le même qu'en simulation. Le choix de la méthode de suivi est primordial au risque de dériver de façon importante par rapport à la position de référence, surtout après une succession d'asservissements visuels qui aura pour effet de cumuler les erreurs de positionnement des caméras. Le manque de robustesse du suivi de points par la méthode KLT sur les images sous-marines, ne nous a pas permis de faire l'acquisition d'images sur une trajectoire effectuée par asservissement durant les expérimentations en mer. Cependant l'acquisition par programmes robots nous a permis d'avoir les images voulues le long de trajectoires prédéfinies. Ceci nous permet de tester la méthode reconstruction 3D en se basant sur des images acquises à intervalles réguliers correspondants à la géométrie de la tête stéréo, ce qui constitue l'hypothèse de base pour notre méthode reconstruction.

Chapitre 3

La reconstruction 3D

La méthode de reconstruction 3D va exploiter les images acquises le long de la trajectoire, définie par la géométrie de la tête stéréo, et réalisée par la méthode d'asservissement visuel décrite précédemment. L'objectif est d'obtenir une reconstruction dense afin de permettre des mesures quantitatives sur l'objet étudié. Le processus de reconstruction 3D se décompose en plusieurs étapes que nous détaillerons dans la première partie. La seconde partie illustrera l'application de cette méthode sur différents exemples dont les images sous-marines.

3.1 Description de la méthode

Le processus de reconstruction 3D peut se décomposer en deux étapes principales qui sont : la reconstruction euclidienne éparse de la structure de l'objet par un nuage de points, la reconstruction dense par un algorithme d'appariement dense, et le rendu visuel obtenu par le plaquage de texture sur un maillage.

3.1.1 La méthode de stéréophotogrammétrie

La méthode de reconstruction choisie s'appuie sur la géométrie épipolaire, et suit un schéma standard dont les principales étapes sont représentées sur la figure 3.1.

L'extraction de points et l'appariement se font exactement comme pour l'asservissement visuel, à la différence que l'on travaille sur une séquence d'images. L'étape de calibration de la paire stéréo doit se faire sur des images prises sur le site, afin de prendre en compte la variabilité des paramètres des caméras. Les paramètres intrinsèques varient en fonction du changement de focale mais également en fonction de l'indice optique de l'eau qui dépend de la température, de salinité, de la pression, et de la longueur d'onde dans l'eau. On utilisera une mire car l'autocalibrage est plus contraignant, et nécessite des mouvements spécifiques de la caméra. Une fois calibrées, on supposera que les paramètres intrinsèques ne varient pas durant toute la phase d'acquisition des images. Une première reconstruction projective est alors réalisée par une méthode d'ajustement de faisceaux (Bundle Adjustment). S'appuyant sur cette reconstruction initiale, un appariement dense des points appartenant à la structure est calculé à partir de paires d'images rectifiées. La rectification des images est également utilisée pour l'appariement des points servant à l'asservissement des caméras. Enfin une carte de profondeur dense est calculée pour chaque point apparié, puis les points 3D sont reliés entre eux par une triangulation de Delaunay. Le plaquage de textures sur les surfaces finit de donner un aspect photo-réaliste à l'objet reconstruit. Cette méthode de reconstruction doit être appliquée sur chaque sous-ensemble de séquence d'images correspondant à chaque station du robot autour de l'objet, puis les différentes parties reconstruites sont alors assemblées grâce à une méthode de recalage sur les modèles partiels 3D. Les algorithmes de reconstruction, tout comme ceux de reconstruction, s'appuient sur des correspondances de points ou de droites entre les images [Bar03]. Mais la localisation de ces points ou droites dans les images est affectée par un bruit de mesure, influençant la qualité des modèles 3D reconstruits.

3.1.2 Reconstruction euclidienne éparse de la structure

La méthode de reconstruction dépend des informations dont on dispose sur le calibrage de la caméra, et sur les positions de prises de vue qui dépendent de la géométrie de la tête stéréo. On parle de reconstruction euclidienne lorsque l'on connaît les paramètres intrinsèques de la caméra, c'est-à-dire lorsque la transformation entre les coordonnées images des points d'intérêts et leurs coordonnées 3D est connue. Cette reconstruction est la plus complète, du fait que la géométrie euclidienne possède les notions de conservation d'angle et de métrique (cf. figure 1.7 page 29). Il est alors possible d'obtenir un modèle métrique de la scène 3D.

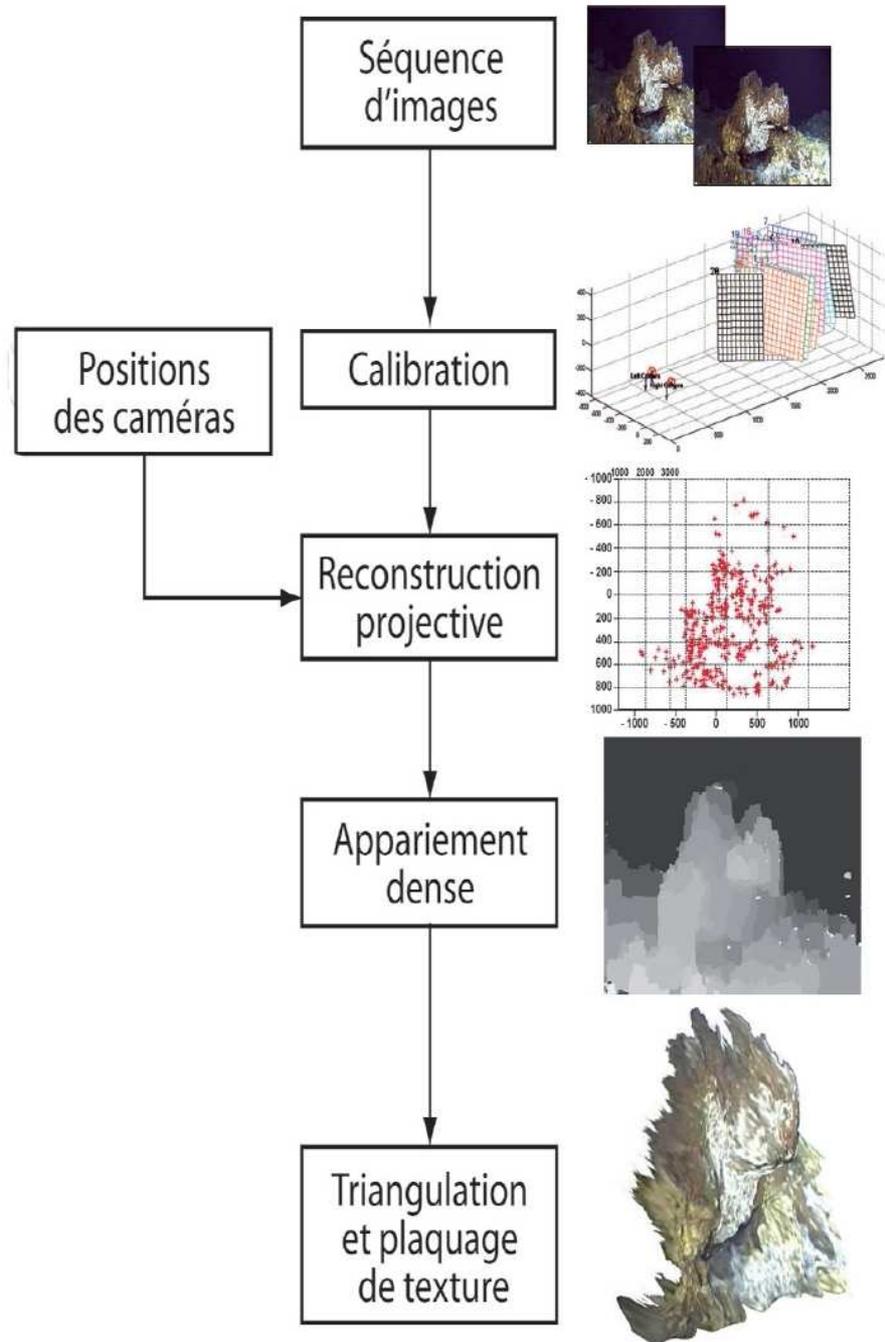


FIG. 3.1 – Méthode générale de reconstruction 3D

Le calcul des coordonnées des points dans l'espace 3D à partir de leurs projections dans les plans images nécessite de connaître les paramètres extrinsèques (positions des prises de vue) et les paramètres intrinsèques des caméras (modèle de calibration de chaque caméra). La précision *in fine* de la reconstruction 3D métrique d'un objet est donc fortement conditionnée par la qualité d'estimation de ces paramètres. Durant les expérimentations d'asservissement visuel utilisant la commande invariante, il n'était pas nécessaire de connaître les paramètres intrinsèques. Ceux-ci sont aisément déduits de la phase de calibration et sont considérés comme constants sur la durée de l'expérimentation tandis que les paramètres extrinsèques représentent les positions des prises de vue exprimées dans un même repère de référence. Chaque position est caractérisée par six paramètres (trois pour la translation et trois pour la rotation). S'appuyant sur le postulat qu'une réduction du nombre d'inconnues intervenant dans le calcul de la reconstruction permet d'aboutir à une reconstruction plus précise et plus stable, notre méthode de reconstruction 3D exploite la contrainte imposée sur le positionnement des caméras par la tâche d'asservissement visuel décrite dans le chapitre précédent. En effet, de par la méthode d'acquisition des images mise en oeuvre, l'espacement des prises de vues est directement lié à la géométrie du système qui est estimée par calibration. Les paramètres des caméras sont calculés hors ligne à partir des paires d'images représentant la mire en utilisant le programme de calibration développée par Jean-Yves Bouguet [BP98]. Les paramètres intrinsèques sont tout d'abord estimés pour chaque caméra, puis les paramètres extrinsèques sont identifiés afin de déterminer la géométrie de la tête stéréo. Une fois cette géométrie déterminée, il devient possible de calculer les positions théoriques des prises de vues autour de l'objet.

3.1.2.1 Extraction et appariement de points

Une des étapes clés de la reconstruction 3D concerne la mise en correspondance des primitives dans les images. La méthode SIFT est utilisée pour extraire un nombre limité de points d'intérêt robustes et invariants aux changements d'échelle, aux rotations et partiellement aux variations d'éclairage. Ces points sont appariés dans chaque paire d'images par corrélation et un algorithme de RANSAC élimine les faux appariements. Peu de mauvais appariements restent après la mise en correspondance, mais ils peuvent être éliminés grâce à la contrainte épipolaire, en estimant la matrice fondamentale comme présenté précédemment. Cette même matrice permettra par la suite de rectifier les images afin de réaliser un appariement dense. Elle est estimée par l'algorithme des 8 points [HZ00], combiné avec un algorithme de RANSAC.

3.1.2.2 Calcul de la structure 3D éparsée par ajustement de faisceaux

A partir de là, connaissant les modèles de projection des caméras et la géométrie d'acquisition des images, il est possible de calculer par triangulation les coordonnées 3D des points appariés. On obtient ainsi une première structure 3D définie par un nuage de points. Afin de compenser les erreurs inévitables lors du déplacement du bras, la solution obtenue est affinée par un algorithme de minimisation, du type ajustement de faisceaux [TMHF00] (Bundle Adjustment). Il consiste à minimiser une fonction de coût représentant la somme des erreurs de reprojection de tous les points. Cet algorithme requiert donc les matrices

de projection \mathbf{P}_k (positions théoriques des caméras et paramètres intrinsèques), et les points 3D χ_i estimés par triangulation. Cette minimisation permet également d'affiner l'estimation des paramètres extrinsèques préalablement au calcul de la reconstruction 3D dense. Les points 3D et les paramètres des caméras optimaux sont trouvés tels que la moyenne du carré des distances entre les points observés sur les images, et les points 3D reprojétés sur les images soit minimisée :

$$\min_{P_k, \chi_i} \sum_{k=1}^m \sum_{i=1}^n D(\mathbf{m}_{ki}, \mathbf{P}_k(\chi_i))^2 \quad (3.1)$$

où $D(\mathbf{m}, \hat{\mathbf{m}})$ représente la distance Euclidienne dans l'image. A ce stade, la structure obtenue nous permet d'avoir les grandes lignes de l'objet, mais la résolution n'est pas suffisante pour avoir une visualisation réaliste permettant des mesures précises.

3.1.3 Reconstruction dense

La structure 3D obtenue précédemment contient seulement un ensemble épars de points 3D. Afin d'obtenir une reconstruction 3D fidèle à la réalité, la structure 3D doit être améliorée par une estimation dense des profondeurs. Cette étape est composée de deux parties principales qui sont la rectification des images suivie de l'appariement dense.

3.1.3.1 Rectification des images

Dans le but de simplifier la procédure d'appariement dense, les images stéréo sont tout d'abord rectifiées. La rectification consiste à transformer deux images dans une géométrie standard de telle manière que les deux plans images soient coplanaires, et que les épipoles soient projetés à l'infini. Pour rectifier les images, la méthode générale présentée dans [Ora01] est utilisée. L'algorithme a seulement besoin des deux images, de la matrice fondamentale précédemment estimée, et de l'ensemble des points SIFT appariés qui ont été utilisés pour calculer la matrice fondamentale. L'homographie permettant de minimiser la déformation entre les images due aux effets de perspective est sélectionnée, puis appliquée sur une image afin de faire coïncider toutes les lignes épipolaires des deux images, c'est-à-dire $\mathbf{H}\mathbf{e}' \simeq \mathbf{e}$.

Etant donné une matrice fondamentale \mathbf{F} , l'ensemble des homographies qui concorde avec la géométrie particulière d'une paire d'images est :

$$\mathbf{H} \approx [\mathbf{e}']_{\times} \mathbf{F} - \mathbf{e}'\mathbf{v}^{\top} \quad (3.2)$$

où \mathbf{v} est un vecteur arbitraire tel que $\det(\mathbf{H}) \neq 0$.

En supposant que les points appariés suivent une erreur de distribution gaussienne, une homographie compatible est estimée en minimisant aux moindres carrés le critère suivant pour n points :

$$\min \sum_{i=1}^n D^2(\mathbf{m}_i, ([\mathbf{e}']_{\times} \mathbf{F} - \mathbf{e}'\mathbf{v}^{\top})\mathbf{m}'_i) \quad (3.3)$$

où l'homographie est paramétrisée comme dans l'équation (3.2). Ensuite, les deux images sont paramétrisées avec les coordonnées polaires centrées sur l'épipôle, pour mettre les lignes épipolaires parallèles à l'axe des images. Ainsi, étant donné un point dans une image, son point correspondant dans la seconde image sera recherché sur la ligne épipolaire horizontale [KPG98].

3.1.3.2 Appariement dense

La seconde étape est de calculer un appariement dense à partir des images rectifiées. Un grand nombre d'algorithmes d'appariement existent et peuvent être classés en deux catégories : les méthodes locales, et les méthodes globales, suivant le principe sur lequel elles sont basées [SS02]. D'autres méthodes appelées algorithmes coopératifs, utilisent l'approche locale et globale en même temps. La difficulté est de choisir un algorithme de reconstruction 3D dense conciliant le rendu visuel, la qualité métrologique, la vitesse de calcul et la complexité de la scène. Le lecteur intéressé trouvera dans [SS02], une comparaison des principaux algorithmes fournissant des informations sur les performances globales (les régions sans texture, les discontinuités de profondeur, les zones d'occlusion). Finalement, la méthode de coupure de graphe (*graph-cut*) donne d'excellents résultats, se montrant meilleure dans les zones sans texture et près des discontinuités, et surpassant les autres méthodes d'optimisation. Les inconvénients majeurs sont : le temps de calcul relativement élevé, et la nécessité d'avoir un réglage précis des paramètres, dont les valeurs sont souvent dépendantes de l'image. Cet algorithme représente cependant un choix très intéressant pour notre application, puisque la qualité du rendu est plus importante que le temps de calcul.

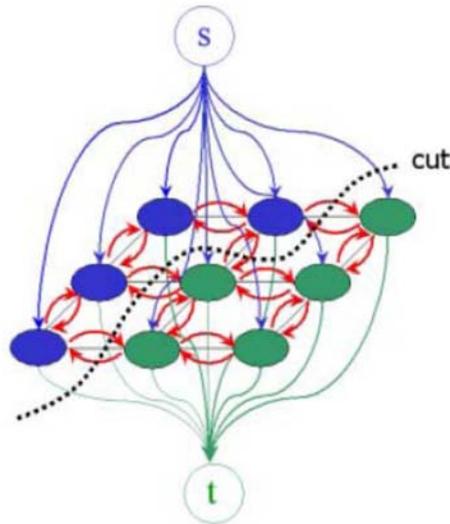


FIG. 3.2 – Coupure d'un graphe sur une image (3×3) avec deux terminaux

Roy and I. J. Cox [RC98] sont les premiers à utiliser cet algorithme dans le contexte de la stéréovision. Afin de présenter le principe de la méthode par coupure de graphe,

nous prenons le cas simple d'un graphe avec seulement deux terminaux tel que sur la figure 3.2. Ce graphe peut être utilisé pour minimiser une fonction d'énergie sur une image de dimensions (3×3) avec deux étiquettes (*labels*). Les deux terminaux sont habituellement appelés la source s et le puit t . Ils correspondent à l'ensemble des étiquettes (les profondeurs des points dans notre cas) qui peuvent être assignés aux pixels. Les différents nœuds représentent les pixels de l'image. Dans le cas général de la théorie de la coupure de graphe, le but est de trouver la coupure qui a un coût minimum parmi toutes les coupures possibles, en minimisant une fonction d'énergie. Soit la fonction f représentant la fonction de disparité associée à chaque pixel d'une image. Nous cherchons l'étiquetage f qui minimise l'énergie. Pour définir la fonction d'énergie pour f , une fonction de coût est introduite basée sur le critère de photoconsistance (similarité entre l'intensité d'un pixel \mathbf{p} de la première image, et le pixel $(\mathbf{p} + f_p)$ de la seconde image) appelé le terme de données (*data term*). Un deuxième terme, appelé terme de lissage (*smoothness term*), pénalise les discontinuités entre pixels voisins. Ainsi, la fonction d'énergie s'écrit :

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{\{p,q\} \in \mathcal{N}} V_{\{p,q\}}(f_p, f_q) \quad (3.4)$$

Où le terme D_p est le terme des données et $V_{\{p,q\}}$ et le terme pénalisant l'uniformité entre pixels adjacents. Dans [KZG03], la minimisation d'énergie considère les images en entrée de façon symétrique, gère correctement la visibilité, et impose une uniformité spatiale (lissage) tout en préservant les discontinuités.

3.1.3.3 Carte de profondeur dense

La carte de disparité donne un appariement dense entre les images stéréo. Ainsi, la carte de profondeur est calculée par triangulation à partir des paires de points appariés, et des paramètres des caméras. Mais chaque point 3D étant considéré indépendamment, il faut lisser la surface afin d'obtenir une cohérence spatiale. Une solution pour une surface continue est d'interpoler la carte de profondeur à l'aide d'un modèle de surface paramétrique [KPG98].

3.1.4 Rendu du modèle 3D

3.1.4.1 Triangulation

La carte de profondeur est convertie en un maillage de surfaces triangulaires en utilisant l'algorithme de triangulation de Delaunay. Ainsi, un plaquage de texture peut être facilement et rapidement appliqué sur la surface de l'objet. Le maillage permet de réduire la complexité géométrique de la représentation 3D, et il permet également de combler quelques vides éventuels sur la surface dus par exemple à des occultations. D'autres techniques existent pour générer des maillages polygonaux, telles que les méthodes frontales, les méthodes par décomposition spatiale hiérarchique en quadtrees-octrees, et les maillages simplex.

3.1.4.2 Plaquage de texture

La représentation 3D est ensuite visualisée avec une apparence plus réaliste en appliquant sur la structure en fil de fer une texture. Tout d'abord, une image de référence est choisie comme texture dans la séquence d'image, puis chaque élément triangulaire de la surface de l'objet est facilement recouvert d'un morceau de l'image, puisque la position de la caméra et la position de la structure 3D sont connues.

3.2 Résultats expérimentaux

Cette partie présente les résultats de reconstruction 3D obtenus avec la méthode détaillée précédemment, pour quatre scènes différentes dont une sous-marine. A la fin de cette partie la précision de la reconstruction est évaluée sur l'un des exemples présentés.

3.2.1 Modélisation d'objets en 3D

Dans cette section, chaque étape de la reconstruction est représentée pour chaque objet afin de suivre pas-à-pas le processus conduisant à la structure 3D, et afin de mieux comprendre les difficultés rencontrées.

3.2.1.1 Conditions expérimentales

Les images ont toutes été acquises par un système stéréo, mais avec des caméras identiques ou différentes, dans l'air ou dans l'eau. La reconstruction ne se fait ici qu'à partir de deux images acquises par la même caméra suite à un déplacement du bras par asservissement visuel ou par programme robot. On n'exploite donc que deux images d'une même caméra prises le long d'une trajectoire contrainte par la géométrie de la tête stéréo. Ceci représente un premier résultat avant d'étendre la méthode à toutes les images de la séquence, ce que nous n'avons pas eu le temps de développer mais qui constitue la suite logique de notre travail. Toutes les caméras utilisées ont été calibrées avant l'acquisition des images. Les expérimentations dans l'air avec le robot manipulateur se sont déroulées sur plusieurs jours d'affilée, mais la calibration de la tête stéréo à l'aide d'une mire n'a été faite qu'une seule fois pour chaque changement de caméras ou modification de la géométrie de la tête. Les à-coups du bras manipulateur au démarrage, à l'arrêt, ou parfois pendant l'asservissement, et les interventions de réglage des caméras parfois défectueuses (problème d'ouverture de l'iris), ont pu légèrement modifier les paramètres des caméras, ce qui explique les corrections obtenues sur l'estimation des structures 3D calculées à partir de ces paramètres.

3.2.1.2 Objets de formes géométriques variées

Les images d'objets de formes géométriques présentées figure 3.3, ont été acquises dans l'air par la même caméra au début et à la fin de l'asservissement visuel. La tête stéréo utilisée pour effectuer l'asservissement visuel est composée de deux caméras CCD identiques espacées de 30cm avec une rotation de 10° de la caméra droite. La raison du choix de ces objets n'est pas de se rapprocher des structures sous-marines de type cheminées hydrothermales pouvant être observées sur les dorsales océaniques, mais de pouvoir tester la méthode sur des objets de tailles et de formes différentes. Le second intérêt dans l'utilisation de ces objets est de pouvoir facilement effectuer des mesures sur les objets afin de les comparer avec celles effectuées directement sur la reconstruction 3D. Un fond neutre a été choisi pour concentrer les points extraits dans l'image sur les objets qui nous intéressent.

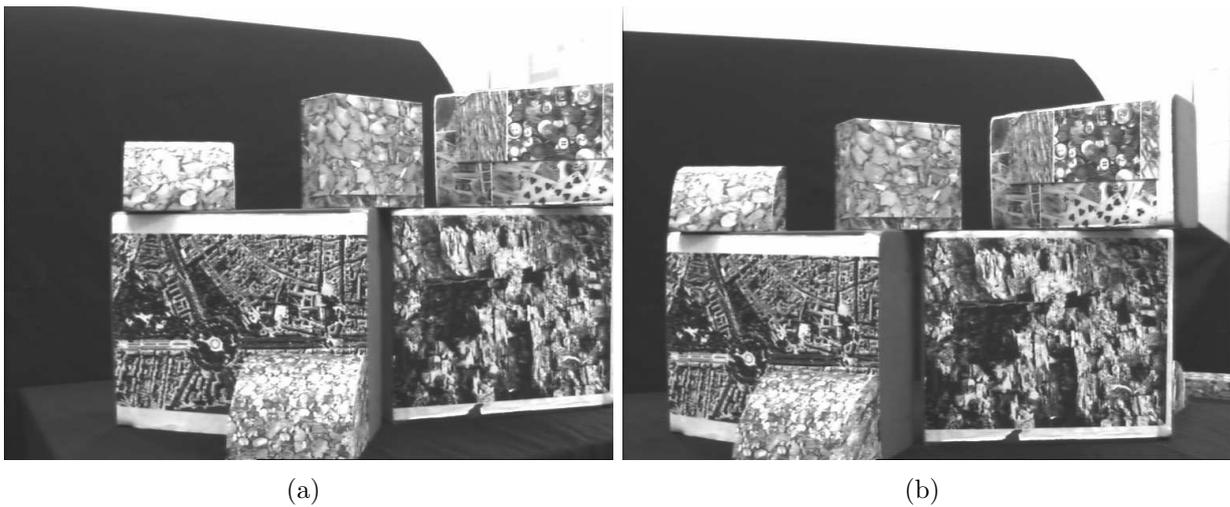


FIG. 3.3 – Images stéréo d’objets de forme géométrique

Une première estimation de la structure est calculée par triangulation à partir de 2289 points SIFT extraits et appariés dans les images stéréo, et des paramètres des caméras obtenus après calibration. La structure 3D ainsi obtenue présentée figure 3.4 permet d’avoir déjà un très bon résultat, grâce entre autre au nombre élevé de points appariés par rapport à la taille des objets, et la simplicité des formes.

Une fonction de minimisation calculée à partir de la somme des erreurs de reprojection des points 3D dans les images permet de corriger la structure 3D ainsi que l’estimation de la position des caméras (cf. §3.1.2.2 page 88). La minimisation est réalisée à l’aide de l’algorithme de Levenberg-Marquardt implémenté par [LA04]. L’algorithme trouve une solution même s’il est initialisé très loin d’un minimum. Chaque itération de Levenberg-Marquardt est une variation de l’itération de Gauss-Newton par l’ajout d’un facteur d’amortissement λ . Ce facteur est ajusté à chaque itération en le divisant par 10 si la convergence est rapide (se rapproche de l’algorithme de celui de Gauss-Newton), ou en le multipliant par 10 si l’itération est peu efficace (se rapproche de l’algorithme de descente de gradient). La fonction arrive à un minimum en 27 itérations, passant d’une erreur initiale de 539,828 à 0,0354741 en 1,31 seconde sur ordinateur équipé d’un processeur Intel P4@3Ghz fonctionnant sous Windows avec 1Go de RAM. La figure 3.5 présente les résultats de cette minimisation sur la structure 3D initiale, calculée par triangulation à partir des points SIFT. La structure corrigée est très proche de la structure initialement calculée. La position des caméras est donnée par 7 paramètres : un quaternion pour la rotation suivi d’un vecteur de 3 éléments pour la translation. Le tableau 3.6 présente la correction de la seconde position de la caméra par rapport à la géométrie de la tête stéréo (la première restant inchangée puisqu’elle est prise comme position initiale). Cette correction est de 9,785mm en translation (distance euclidienne), et de $3,970.10^{-4}^\circ$ en rotation.

Les images stéréo de la figure 3.3 sont ensuite rectifiées à partir des points extraits, afin de réaliser plus facilement l’appariement dense (cf. §3.1.3.1 page 89). Le code utilisé est une implémentation d’une méthode générale de rectification décrite dans [Ora01]. L’angle

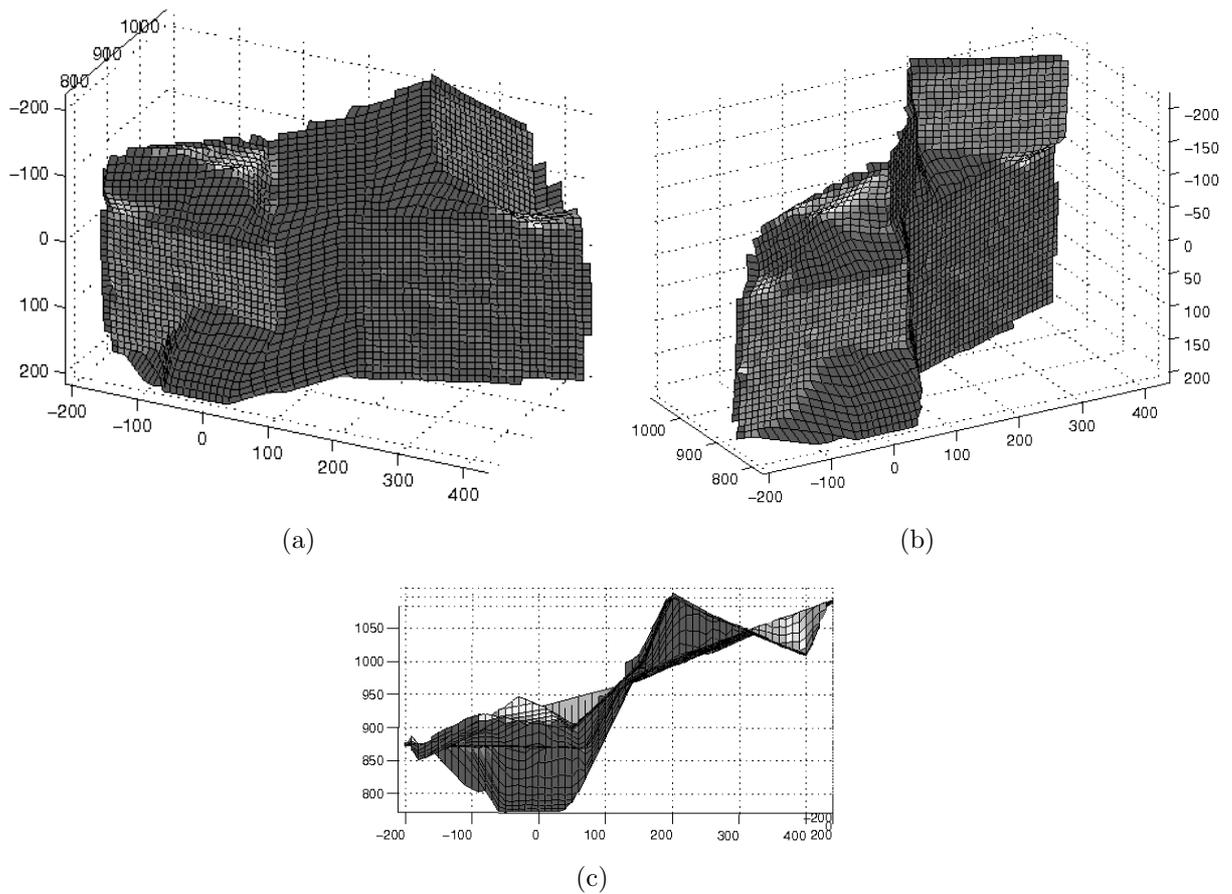


FIG. 3.4 – Estimation de la structure des objets après triangulation

et l'écartement entre les positions des deux vues n'étant pas très élevés (10° et 30cm), la rectification des images présentée figure 3.7 n'est pas très importante.

A partir des images rectifiées, la carte de disparité dense est calculée suivant la méthode de Graph-Cut présentée précédemment (cf. §3.1.3.2 page 90). Un intervalle de disparité est déterminé à partir des points SIFT initiaux. Celui-ci est de $[-1 \ 66]$ en abscisse, et de $[-1 \ 0]$ en ordonnée. La figure 3.8 représente la carte de disparité obtenue. On retrouve les propriétés de la méthode utilisée avec une succession de zones relativement homogènes et des contours assez bien définis entre les objets.

La carte de disparité est ensuite utilisée pour calculer la carte de profondeur dense, avec les paramètres de calibration de la caméra ainsi que les paramètres extrinsèques corrigés précédemment. En rajoutant par plaquage la texture à partir de la première image, on obtient le résultat final de la figure 3.9. La structure 3D présente des déformations (figure 3.9.b et 3.9.d) dès que l'on change de façon importante l'angle de vue initial (figure 3.9.a). En effet le faible nombre d'images limite la quantité d'informations exploitables servant à calculer la structure 3D. Le rapprochement entre les deux vues permet cependant un grand nombre d'appariements mais au détriment de la précision de la triangulation. On peut donc espérer de meilleurs résultats avec un plus grand nombre d'images

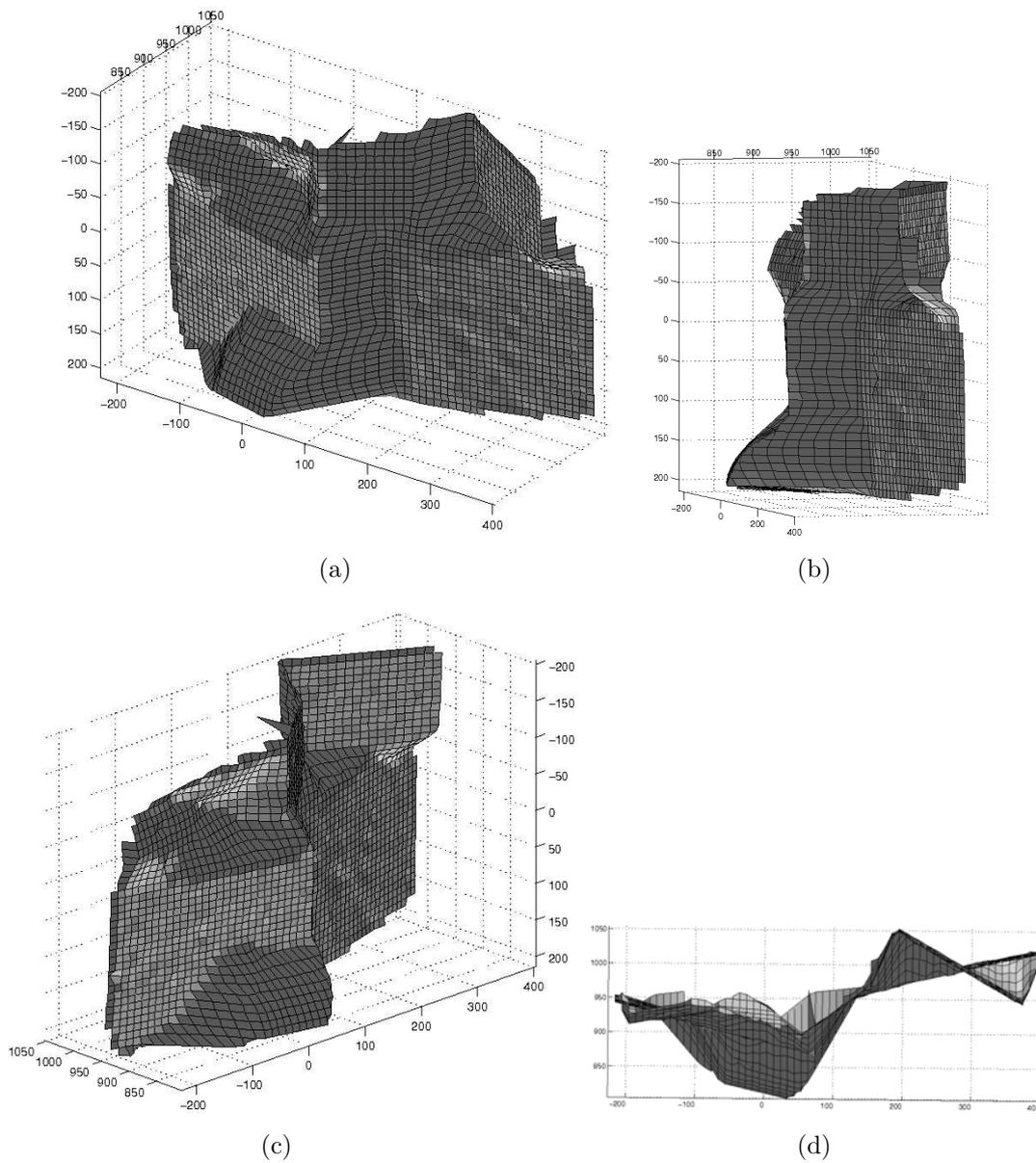


FIG. 3.5 – Estimation de la structure des objets après minimisation par ajustement de faisceaux

couvrant plus d'espace autour de l'objet.

Cam_2	0,996195	0,000000	0,087156	0,000000	-284,37621	0,000000	56,95467
Cam_2'	0,995630	-0,010760	0,095993	0,006162	-284,626131	3,999336	48,028331

FIG. 3.6 – Correction par ajustement de faisceaux de l'estimation de la position de la caméra gauche après un asservissement visuel (1 quaternion (1×4) correspondant à la rotation, et un vecteur de translation (1×3))

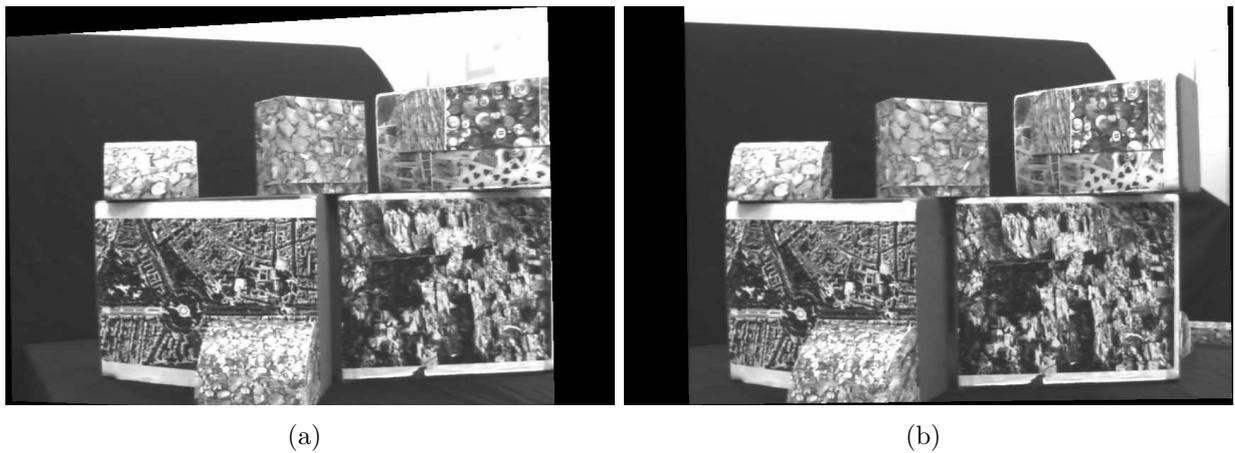


FIG. 3.7 – Images stéréo rectifiées

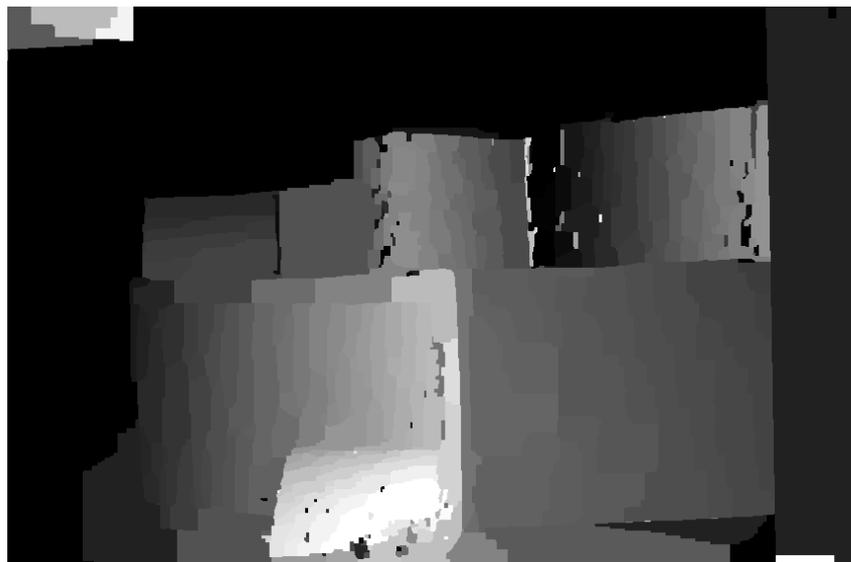


FIG. 3.8 – Carte de disparité dense sur les images rectifiées avec la méthode de Graph-Cut

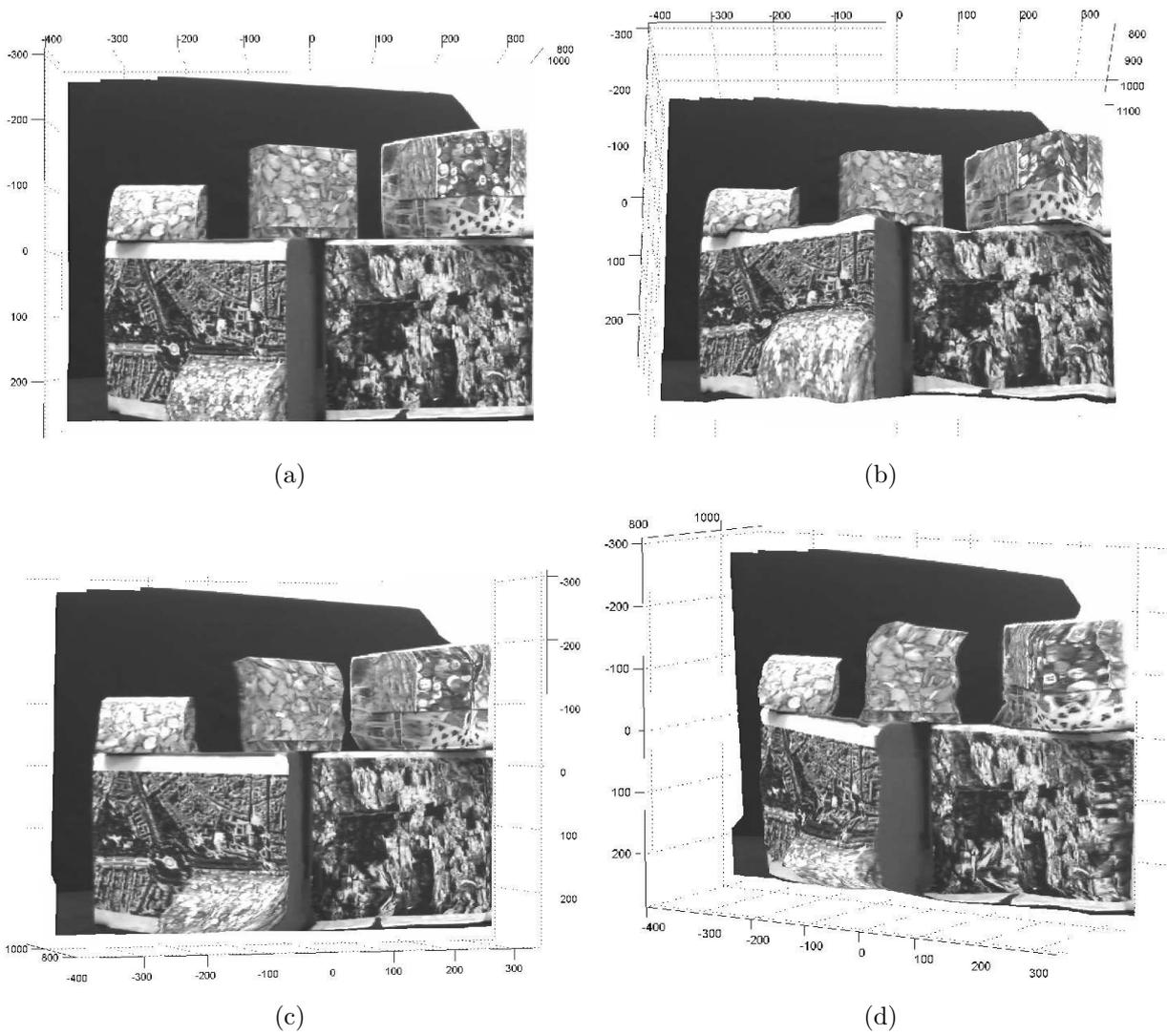


FIG. 3.9 – Structure 3D des objets avec texture

3.2.1.3 Objet cylindrique

Pour se rapprocher des images des cheminées hydrothermales ou des petites structures sous-marines, nous avons utilisé les images de l'objet cylindrique qui a servi à tester l'asservissement visuel dans le chapitre précédent. Dans un premier temps nous avons utilisé un poster plaqué sur la structure pour avoir une surface lisse mais très texturée, puis nous avons recouvert la surface de morceaux de chêne liège pour recréer une surface similaire à celle d'un rocher. Les images stéréo sont issues de la même caméra après un déplacement de la tête stéréo par un programme robot. La géométrie de la tête est de 15° sur la caméra droite et de 20 cm entre les caméras.

- Cylindre recouvert d'un poster



FIG. 3.10 – Images stéréo d'un cylindre recouvert d'un poster

Une première estimation de la structure est calculée par triangulation à partir de 649 points SIFT extraits et appariés dans les images stéréo (figure 3.10), et des paramètres des caméras obtenus après calibration. La structure 3D ainsi obtenue présentée figure 3.11, donne déjà un très bon aperçu de la surface lisse de l'objet de forme cylindrique.

Afin de corriger la structure 3D obtenue précédemment, nous appliquons à nouveau la méthode de minimisation. La fonction arrive à un minimum en 35 itérations, passant d'une erreur initiale de 125,579 à 0,0206357 en 0,484 seconde sur un ordinateur doté des mêmes capacités que pour l'exemple précédent. La figure 3.12 présente les résultats de cette minimisation sur la structure 3D initiale. La structure corrigée se rapproche d'avantage de l'objet réel par sa forme, la structure semble avoir été redressée à la verticale. Il subsiste une anomalie sur la surface lisse due à un mauvais appariement, mais ce couple de points ne sera pas gardé pour le résultat final, étant donné que les points SIFT servent uniquement à corriger les paramètres des caméras et à rectifier les images pour effectuer un appariement dense. Le tableau 3.13

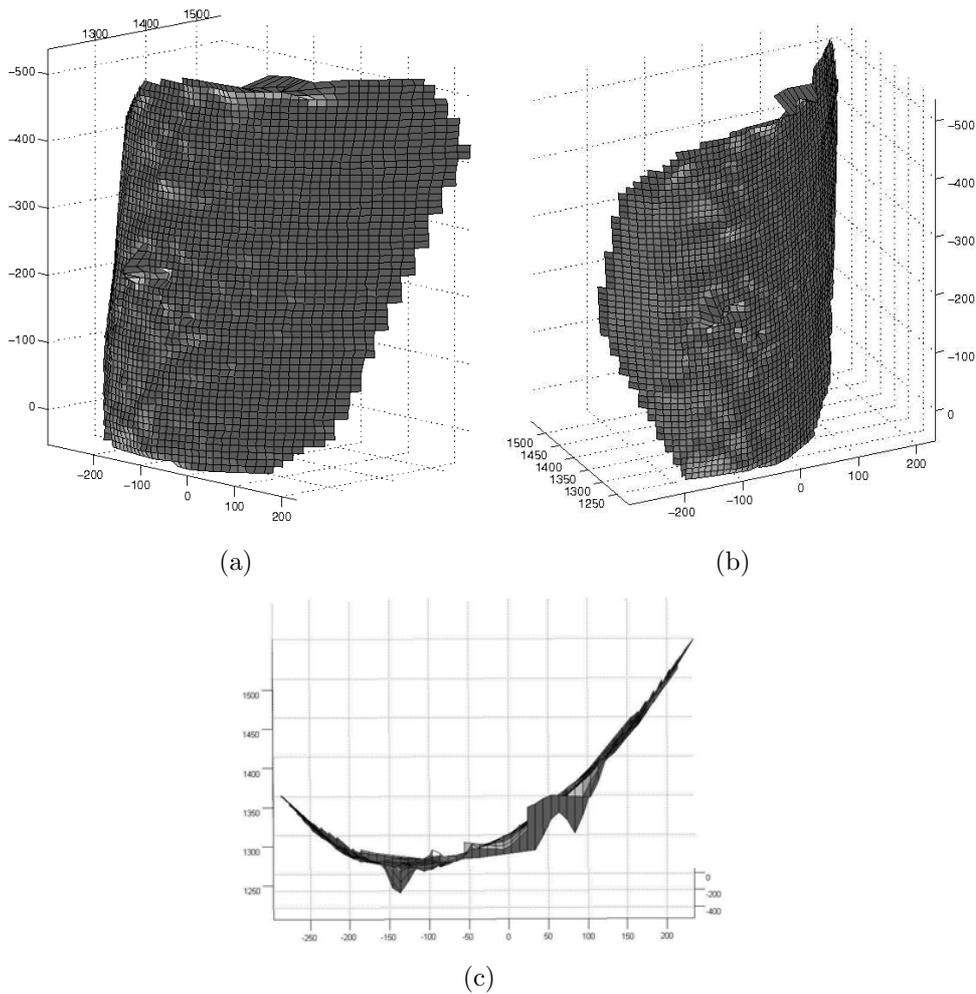


FIG. 3.11 – Estimation d'une partie de la structure après triangulation

présente la correction de la seconde position de la caméra, qui est de 13,661mm en translation, et de $0,907^\circ$ en rotation.

Les images stéréo de la figure 3.10 sont ensuite rectifiées à partir des points SIFT extraits, permettant une mise en correspondance plus aisée. Le résultat de la rectification des images stéréo est présenté sur la figure 3.14.

La carte de disparité dense est ensuite calculée à partir des images rectifiées suivant la même méthode d'appariement dense. L'intervalle de disparité, déterminé à partir des points SIFT initiaux, est de $[-2 \ 72]$ en abscisse, et de $[-1 \ 0]$ en ordonnée. La figure 3.8 représente la carte de disparité obtenue. Le cylindre apparaît nettement au milieu de l'image grâce à des contours bien définis, et un dégradé faisant apparaître une surface lisse et convexe.

La carte de profondeur dense est ensuite obtenue à partir des disparités, des paramètres de calibration de la caméra, et des paramètres extrinsèques corrigés précédemment. La texture est ensuite rajoutée à partir d'une des deux images pour donner le résultat de la figure 3.16. La forme générale est fidèle à l'objet réel, mis à

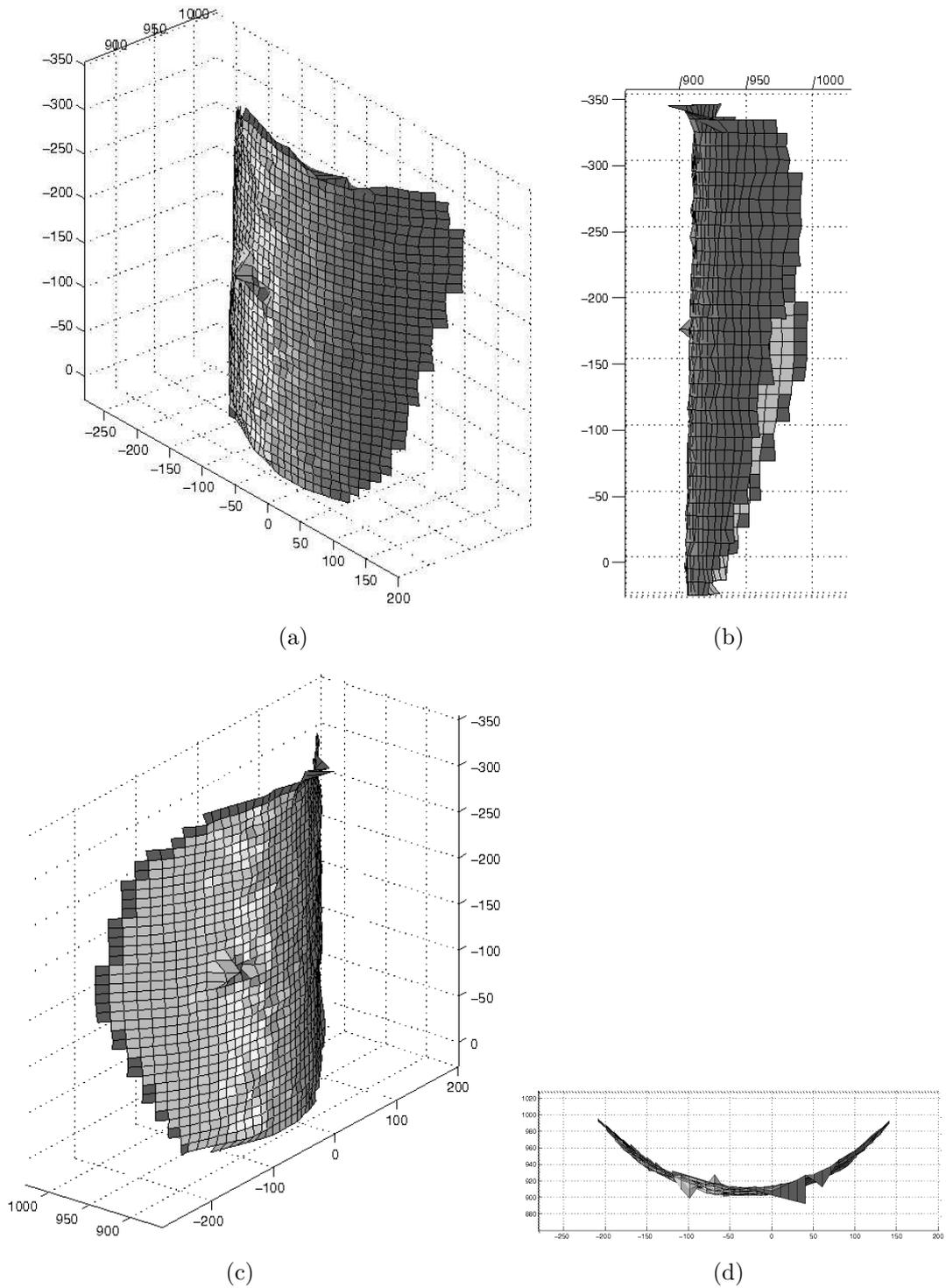


FIG. 3.12 – Estimation d’une partie de la structure après minimisation par ajustement de faisceaux

part quelques déformations locales visibles à certains endroits de la surface comme à la droite de la figure 3.16.d. La forme simple de l’objet permet de grands déplace-

Cam_2	0,991445	0,000000	0,130526	0,000000	-193,185165	0.000000	51.763809
Cam_2'	0,99447	-0,003162	0,129083	0,010470	-204,018982	1,867710	43,654186

FIG. 3.13 – Correction par ajustement de faisceaux de l'estimation des paramètres de la position de la caméra droite après un déplacement par programme robot (1 quaternion (1×4) correspondant à la rotation, et un vecteur de translation (1×3))

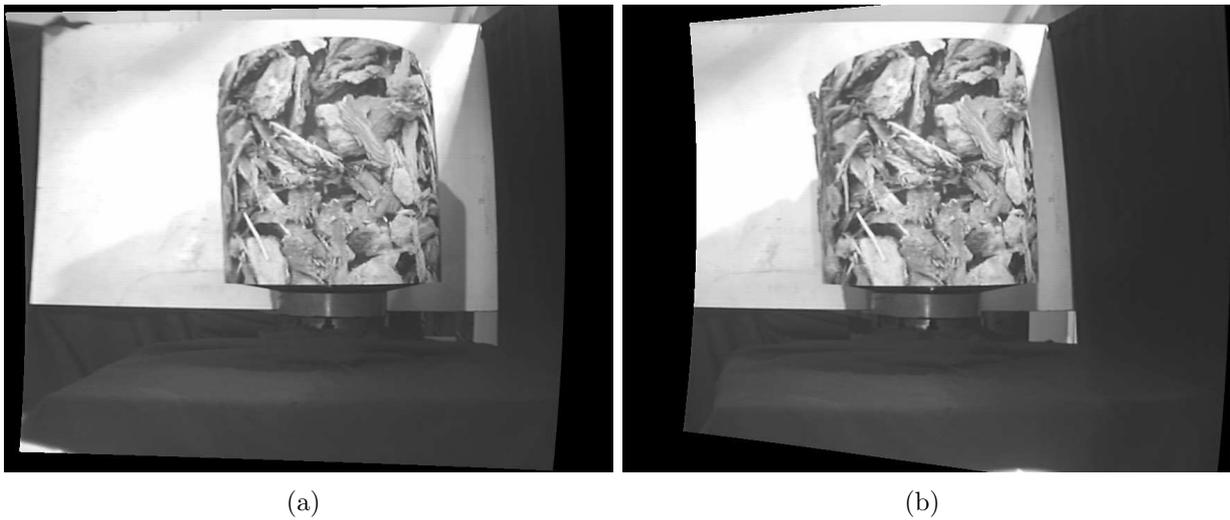


FIG. 3.14 – Images stéréo rectifiées

ments autour de la structure 3D tout en conservant l'aspect cylindrique de l'objet, comme on peut le voir sur la figure 3.16.c représentant une vue de profil.

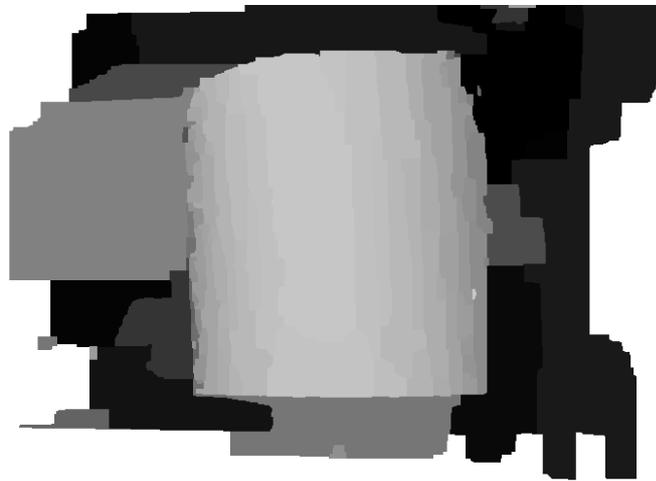


FIG. 3.15 – Carte de disparité dense sur les images rectifiées avec la méthode de Graph-Cut

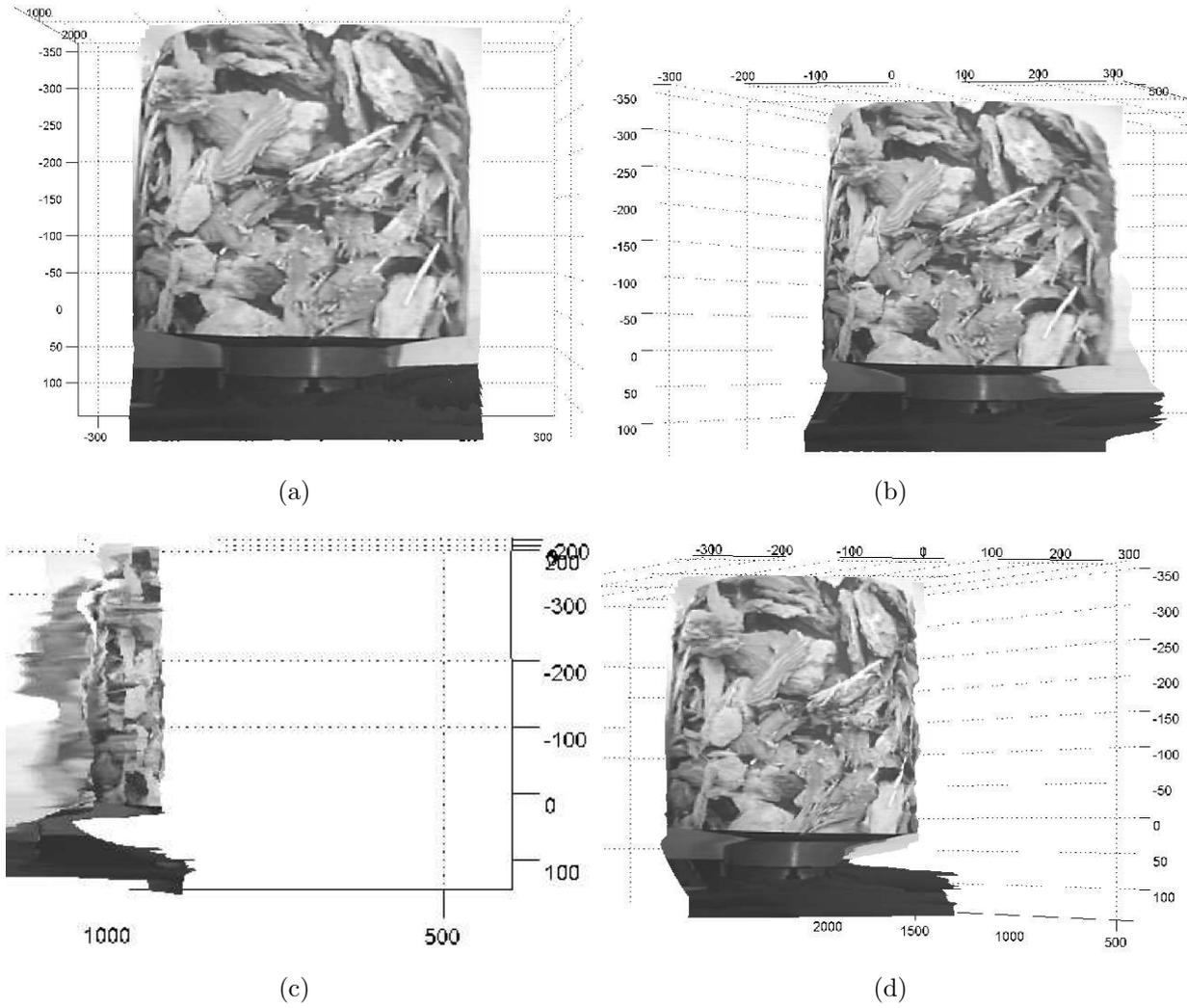


FIG. 3.16 – Structure 3D des objets avec plaquage de texture

– Cylindre recouvert de morceaux d'écorce

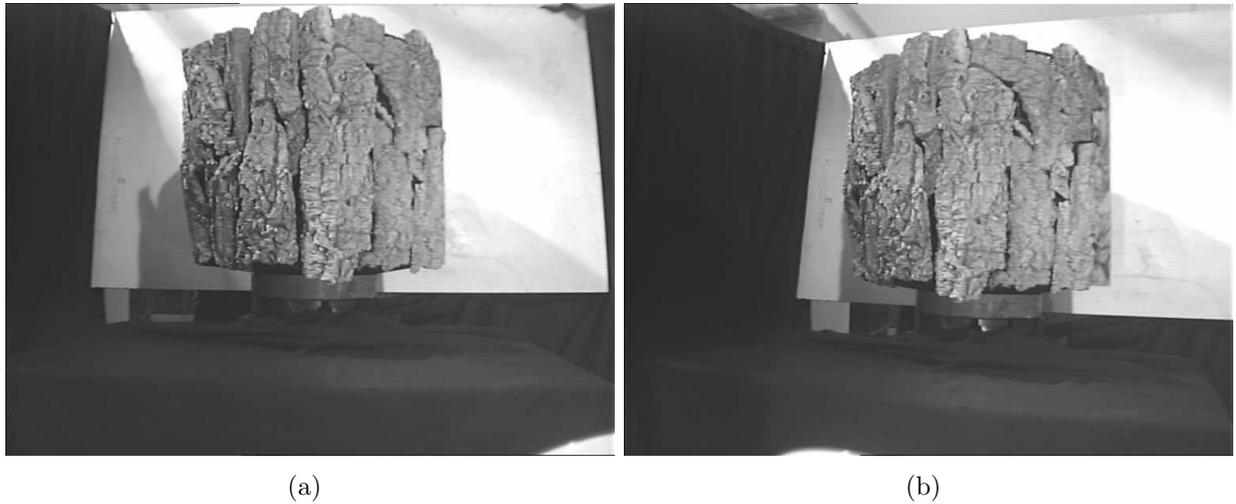


FIG. 3.17 – Images stéréo d'un cylindre recouvert de morceaux d'écorce

L'utilisation d'une texture naturelle que représente l'écorce de chêne liège, nous permet de franchir une étape de plus vers la reconstruction d'objets naturels sous-marins. L'estimation de la structure est calculée par triangulation à partir de 293 points SIFT extraits et appariés dans les images stéréo (figure 3.17), ainsi que des paramètres des caméras calibrées. La structure 3D ainsi obtenue présentée figure 3.18, reste quasiment identique dans sa globalité à la première estimation de la structure du cylindre recouvert d'un poster. La seule différence est la surface beaucoup plus irrégulière qui suit les contours des morceaux d'écorce.

Le résultat de la correction de la structure par la fonction de minimisation est également semblable au résultat de l'objet précédent, mais celui-ci est obtenu avec deux fois moins de points. La fonction arrive à un minimum en 31 itérations, passant d'une erreur initiale de 290,905 à 0,0336566 en 0,230 seconde. La figure 3.19 présente les résultats de cette minimisation sur la structure 3D initiale, avec une nette amélioration sur la vue de dessus de la structure visible sur la figure 3.19.b. Le tableau 3.20 présente la correction de la seconde position de la caméra qui est de 17,129mm en translation, et de 0,554° en rotation, ce qui reste du même ordre de grandeur que précédemment.

Le résultat de la rectification des images stéréo (figure 3.17), calculée à partir des 293 points SIFT, est présenté dans la figure 3.21.

La figure 3.22 représente la carte de disparité obtenue à partir des images rectifiées par la méthode de Graph-Cut. L'intervalle de disparité pour l'appariement des points est de $[-4 \ 77]$ en abscisse, et de $[-13 \ 0]$ en ordonnée. On distingue l'objet au milieu de l'image, avec un dégradé faisant apparaître la surface globalement convexe, avec des contours irréguliers rappelant la texture de l'écorce.

La figure 3.23 représente différentes vues de la structure 3D finale, obtenue après le calcul de la carte de profondeur dense et du plaquage de la texture. Le résultat reste

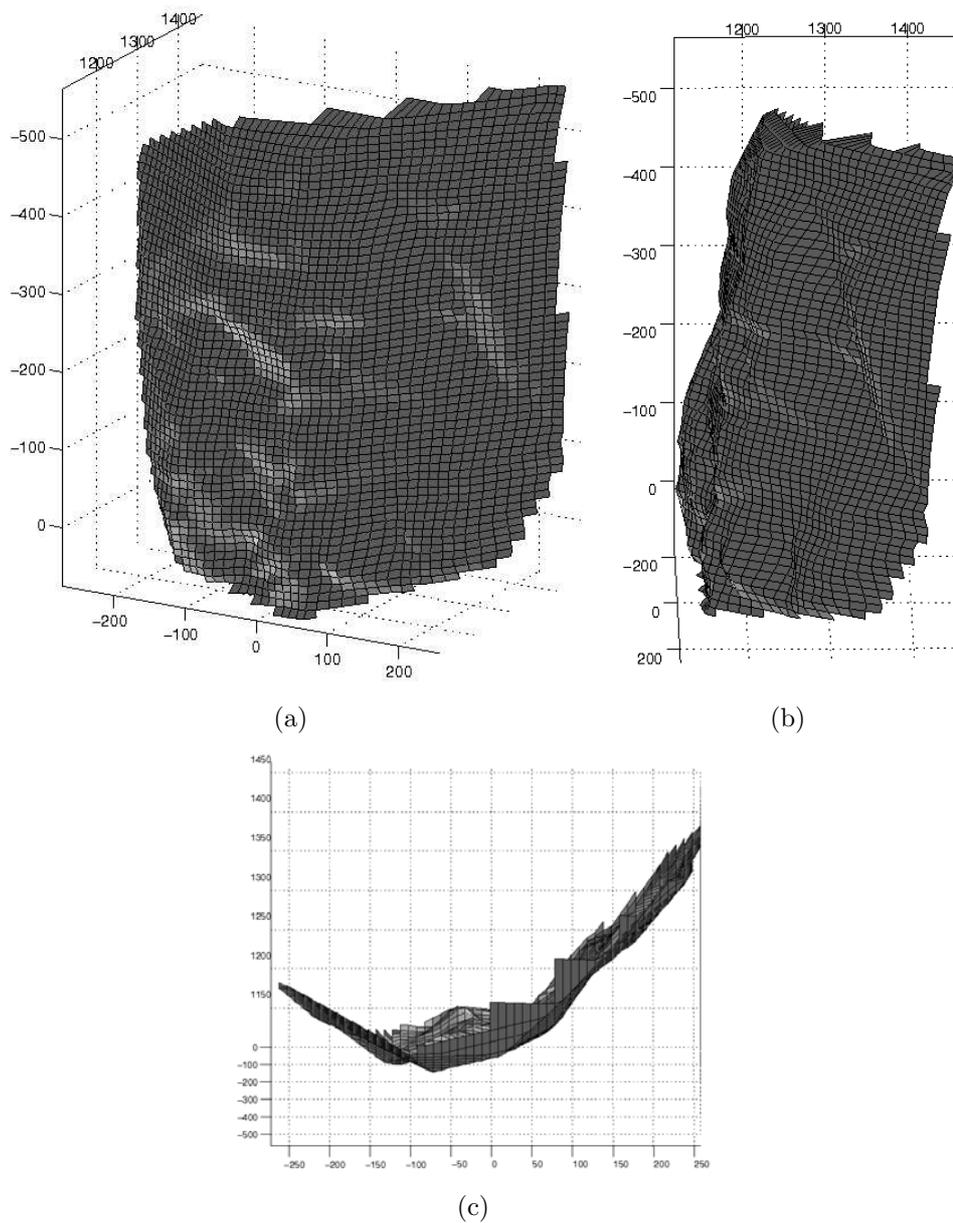


FIG. 3.18 – Estimation d’une partie de la structure de l’objet après triangulation

semblable à la structure 3D du résultat précédent malgré le changement de surface du cylindre, ce qui nous permet d’espérer des résultats satisfaisants sur les images d’une scène naturelle.

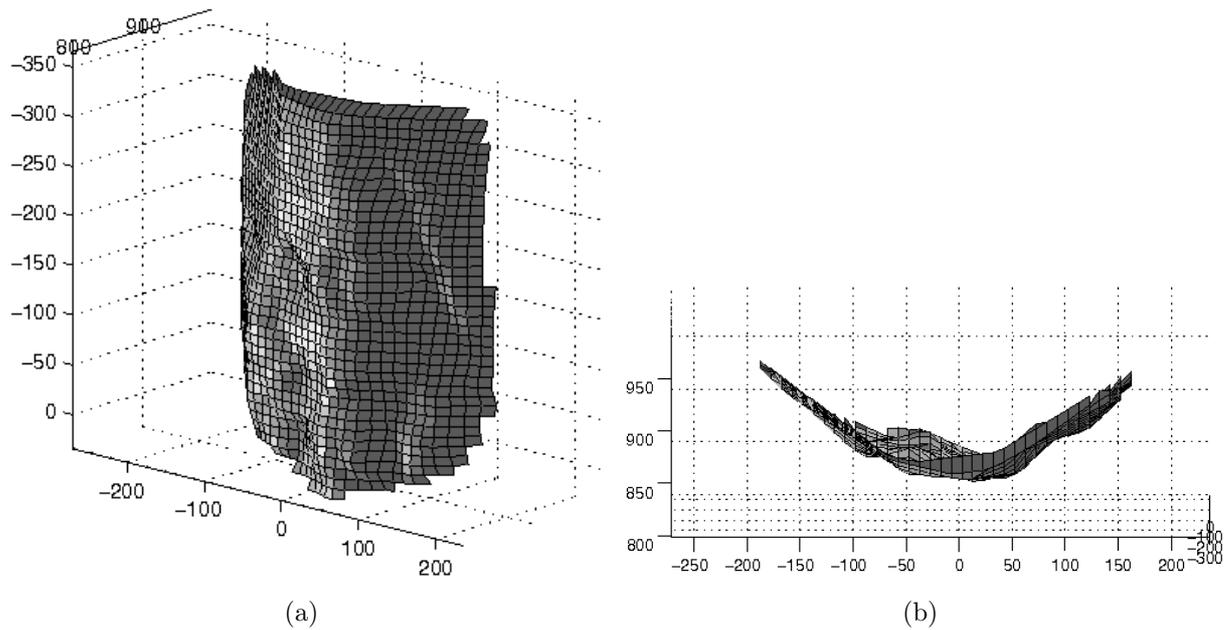


FIG. 3.19 – Estimation d’une partie de la structure de l’objet après minimisation par ajustement de faisceaux

Cam_2	0,991445	0,000000	0,130526	0,000000	-193,185165	0,000000	51,763809
Cam_2	0,992064	-0,013402	0,127173	0,008612	-196,448288	1,784355	35,043226

FIG. 3.20 – Correction par ajustement de faisceaux de l’estimation des paramètres de la position de la caméra droite après un déplacement par programme robot (1 quaternion (1×4) correspondant à la rotation, et un vecteur de translation (1×3))

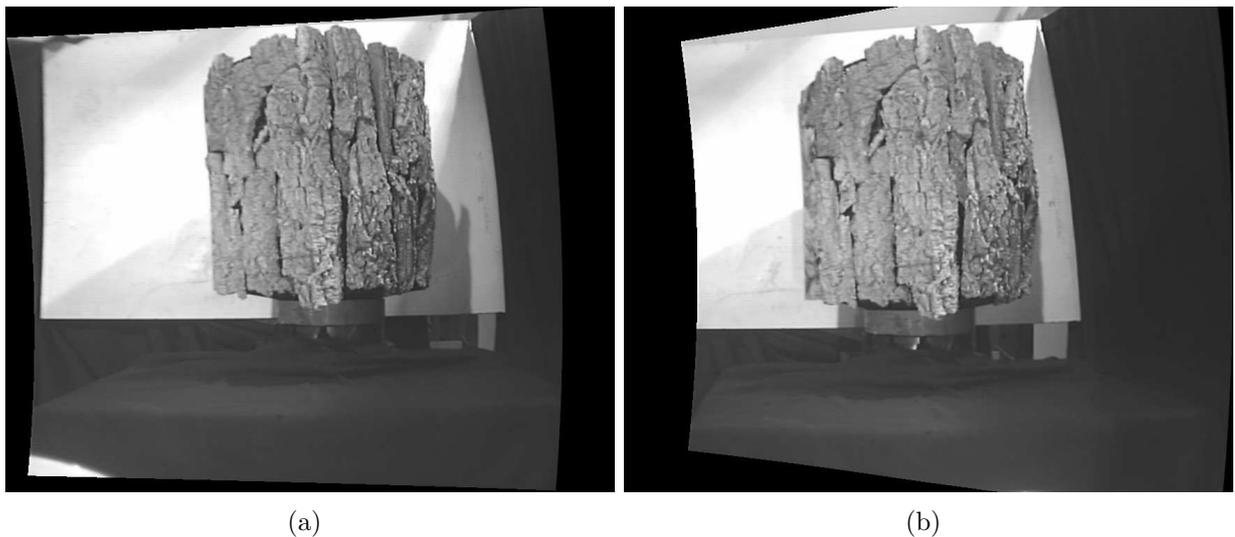


FIG. 3.21 – Images stéréo rectifiées

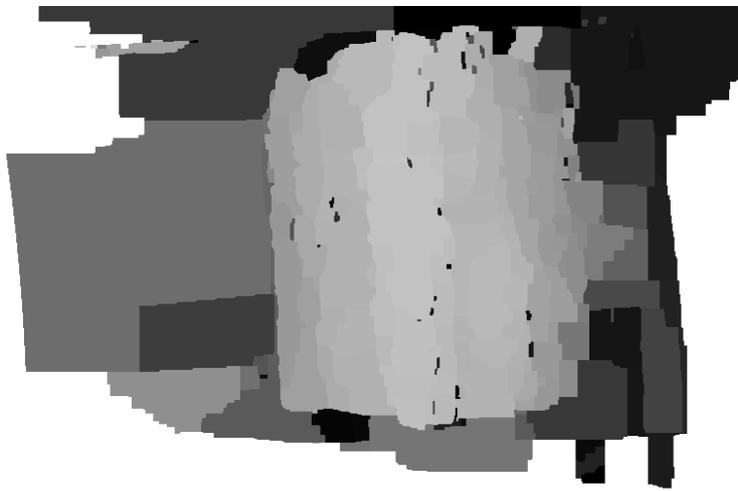
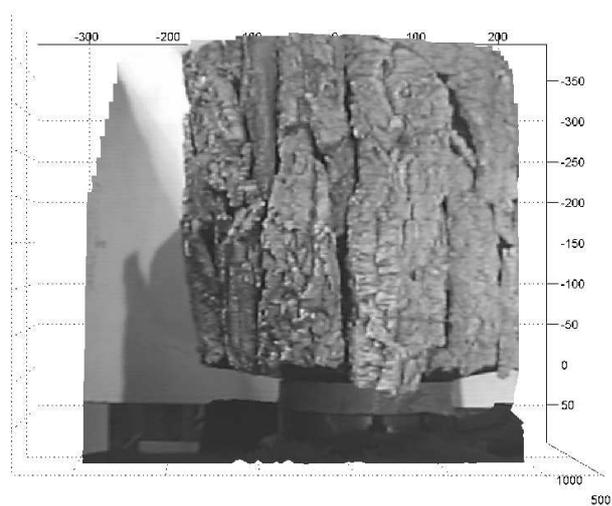
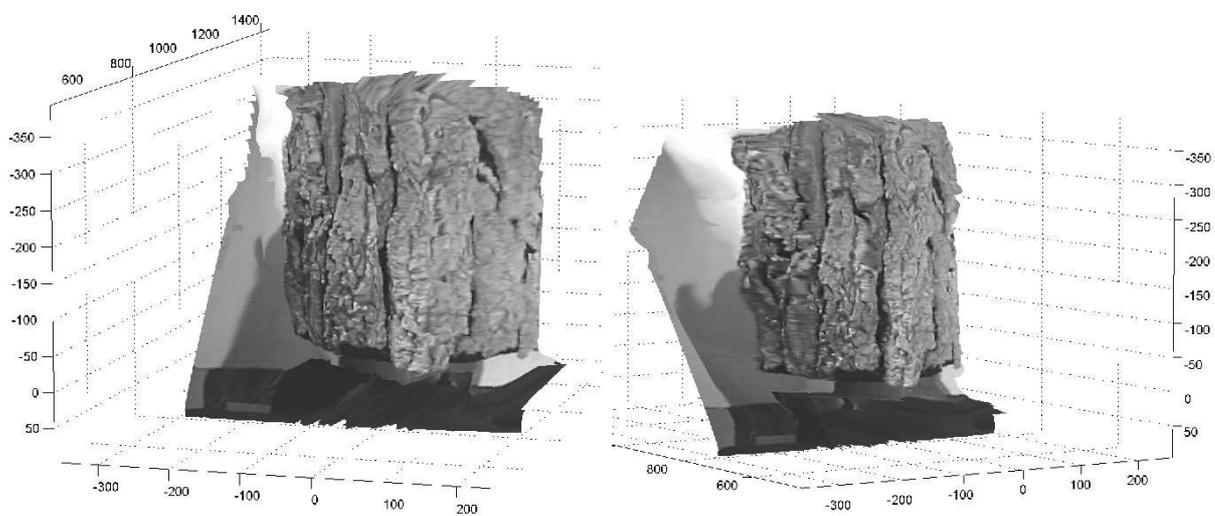


FIG. 3.22 – Carte de disparité dense sur les images rectifiées avec la méthode de Graph-Cut



(a)



(b)

(c)

FIG. 3.23 – Structure 3D de l'écorce avec plaquage de texture

3.2.1.4 Reconstruction d'un objet sous-marin



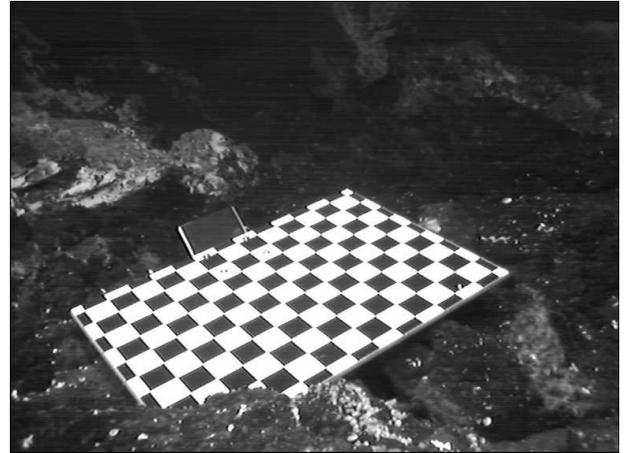
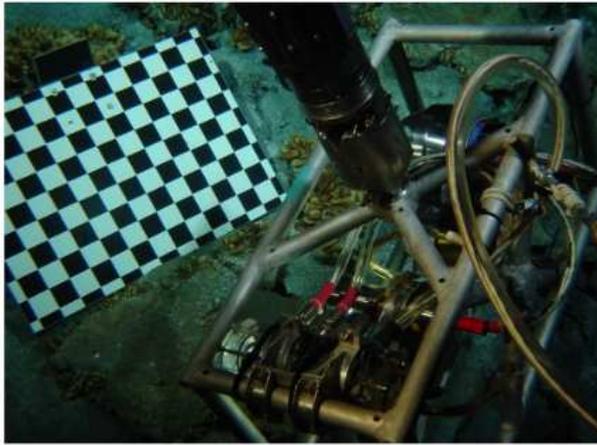
FIG. 3.24 – Images stéréo d'une scène sous-marine

Le dernier exemple de reconstruction 3D permet de valider la méthode sur un objet sous-marin naturel, à partir de l'acquisition d'images stéréo sur une trajectoire effectuée par un bras manipulateur, ce qui constitue l'objectif de ce travail de recherche. La paire d'images sous-marines choisie servant à la reconstruction 3D a été acquise dans les conditions présentées précédemment (cf. §2.3.6 page 77). Une première estimation de la structure est calculée par triangulation à partir de 265 points SIFT extraits et appariés dans les images stéréo (figure 3.26), et des paramètres des caméras obtenus après calibration *in situ* (figure 3.25).

La fonction servant à corriger les points 3D et l'estimation des positions des caméras, arrive à un minimum en 34 itérations, passant d'une erreur initiale de 51,2293 à 0,0824 en 0,188 seconde. La figure 3.27 présente les résultats de cette minimisation sous la forme d'une première reconstruction 3D après une triangulation de Delaunay sur le nuage de points, et un plaquage de texture. La structure 3D est très ressemblante aux grandes lignes de l'objet observé dans les images. Afin d'affiner ce résultat pour obtenir une meilleure précision, un appariement dense est tout de même nécessaire. Le tableau 3.28 présente la correction de la seconde position de la caméra qui est de 25,084mm en translation, et de $0,961^\circ$ en rotation.

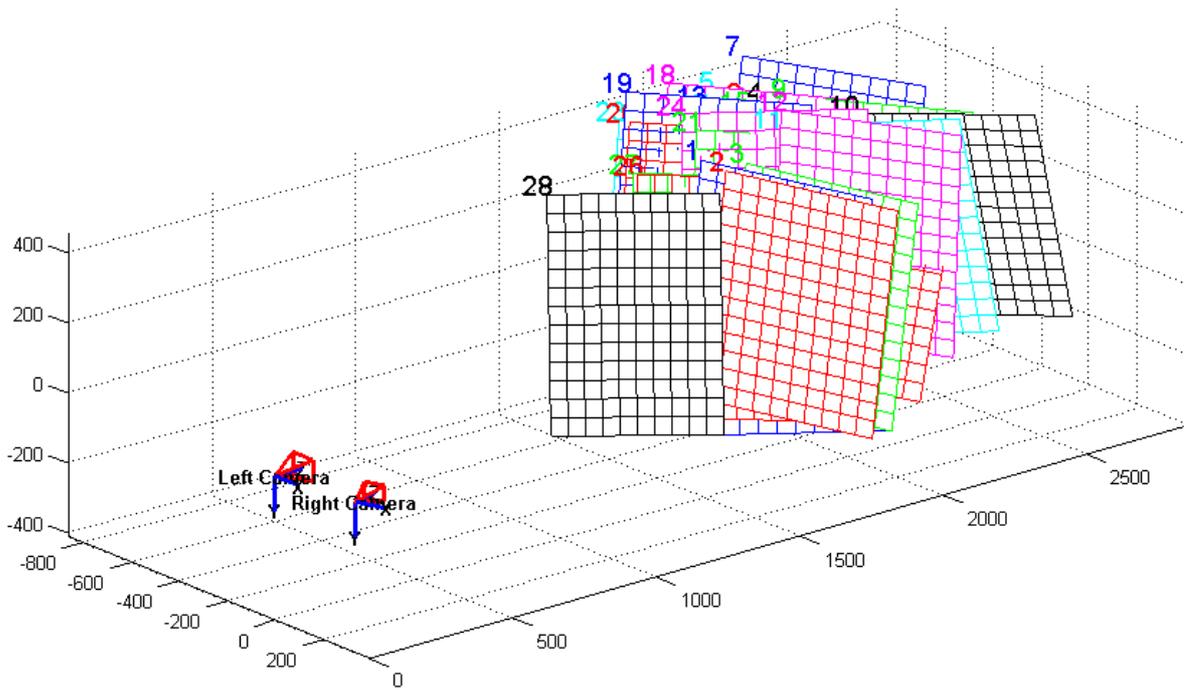
Les images stéréo de la figure 3.24 sont ensuite rectifiées à partir des points extraits (figure 3.29), afin d'obtenir la carte de disparité dense de la figure 3.30. Cette carte de disparité présente bien les différents plans de la scène, avec l'objet à reconstruire devant, puis le sol en dégradé, et enfin le fond en noir.

La figure 3.31 représente le résultat final obtenu à partir de seulement deux images sous-marines. La carte de profondeur sur la figure 3.31.a, permet de mieux se rendre compte du relief de l'objet, tandis que le plaquage de texture sur la figure 3.31.b, donne un aspect réaliste au résultat final tout en gommant visuellement certains défauts de la reconstruction.



(a) Acquisition des images avec la tête stéréo IRIS

(b) Image de la caméra gauche de la mire de calibration



(c) Résultat de la calibration

FIG. 3.25 – Calibration de la tête stéréo IRIS sur le site d'expérimentation

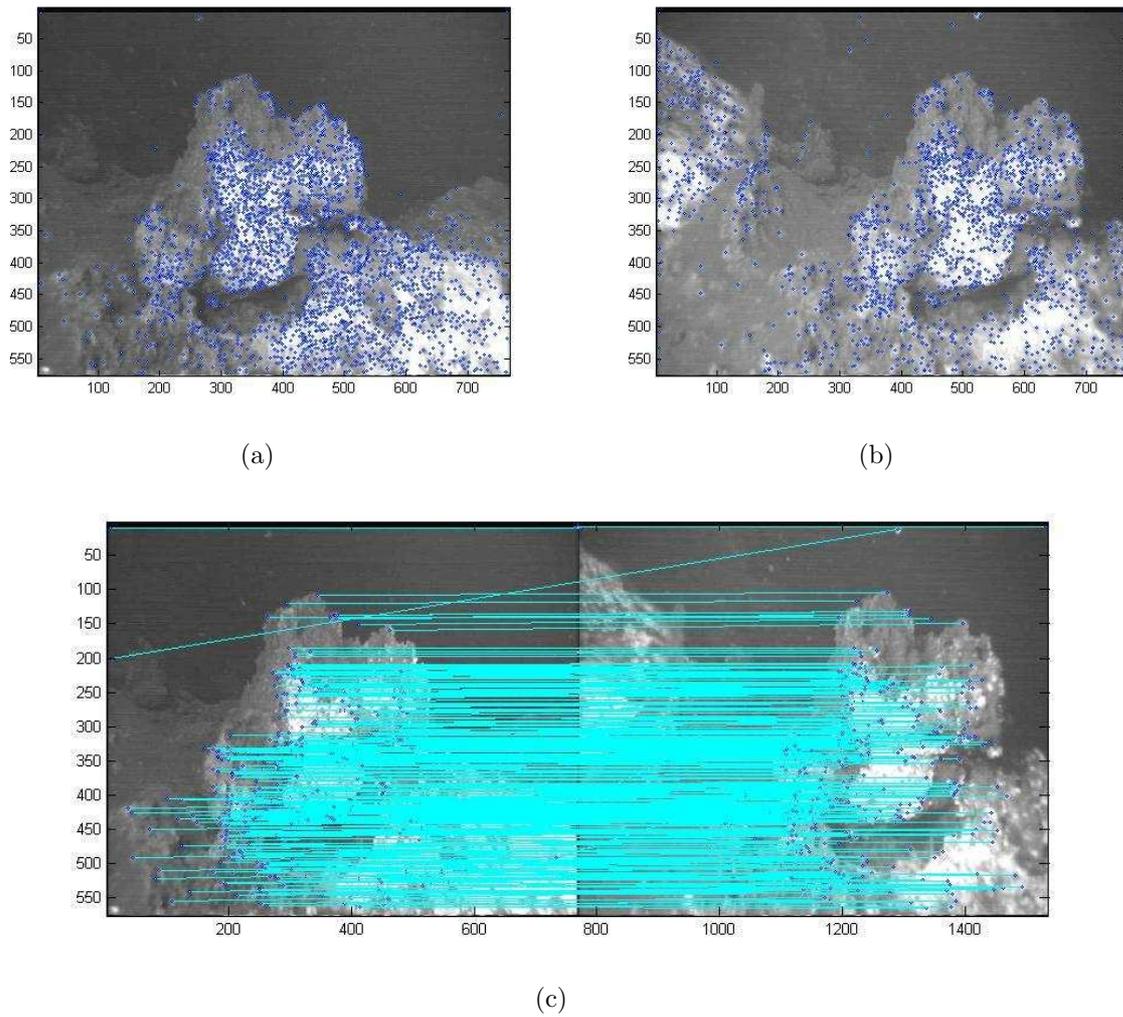
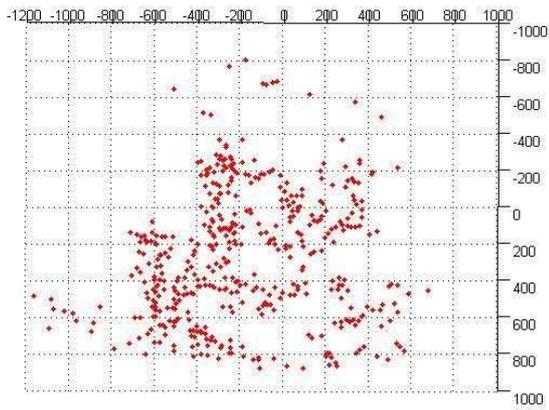
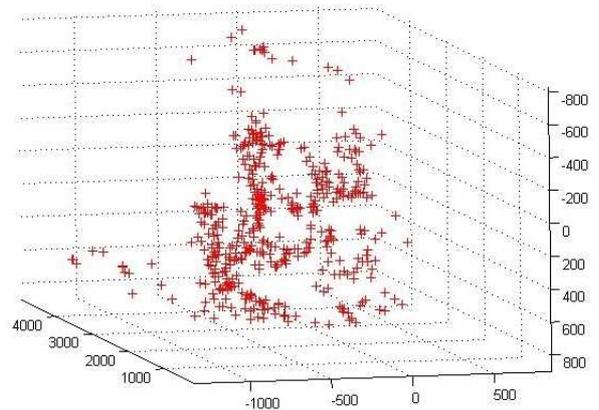


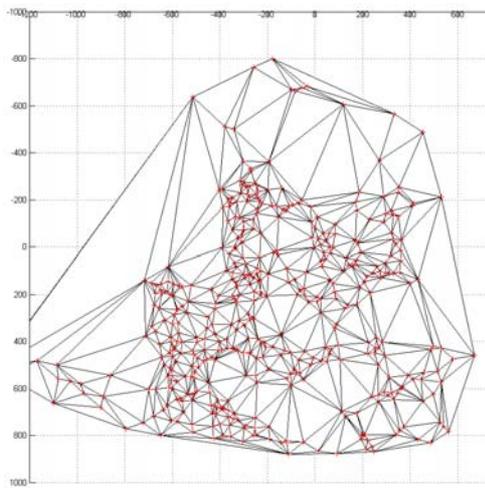
FIG. 3.26 – Extraction et appariement des points SIFT dans les images stéréo



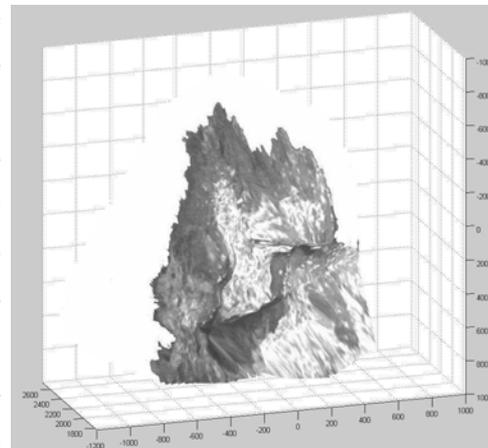
(a) Nuage de points corrigés



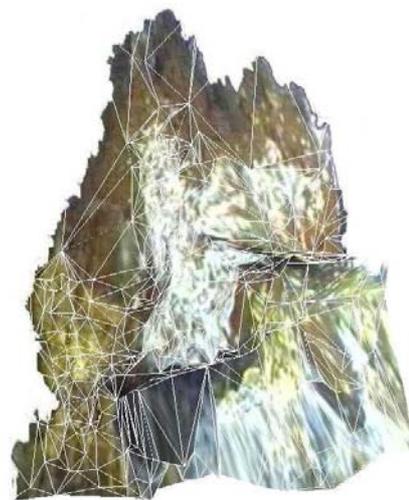
(b) Vue 3D du nuage de points



(c) Triangulation de Delaunay sur les points corrigés



(d) Structure 3D avec texture



(e) Structure 3D à partir des points SIFT

FIG. 3.27 – Correction des points 3D et de l'estimation des positions des caméras

Cam_2	0,991445	0,000000	0,130526	0,000000	-289,777748	0,000000	77,645714
Cam'_2	0,992505	0,007226	0,122230	-0,010564	-274,715490	11,329868	61,094079

FIG. 3.28 – Correction par ajustement de faisceaux de l'estimation des paramètres de la position de la caméra gauche après un déplacement par programme robot (1 quaternion (1×4) correspondant à la rotation, et un vecteur de translation (1×3))

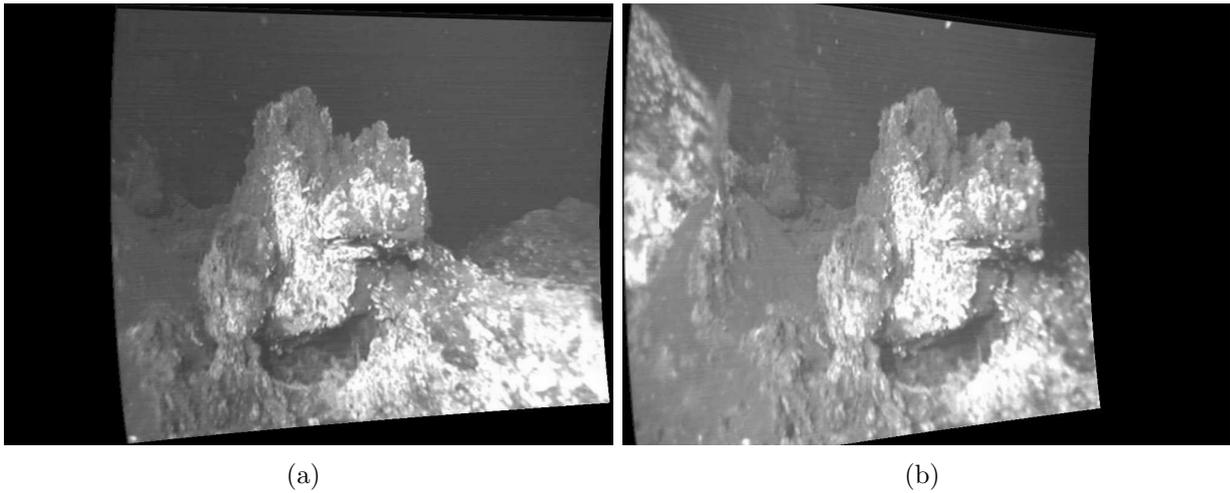


FIG. 3.29 – Images stéréo rectifiées

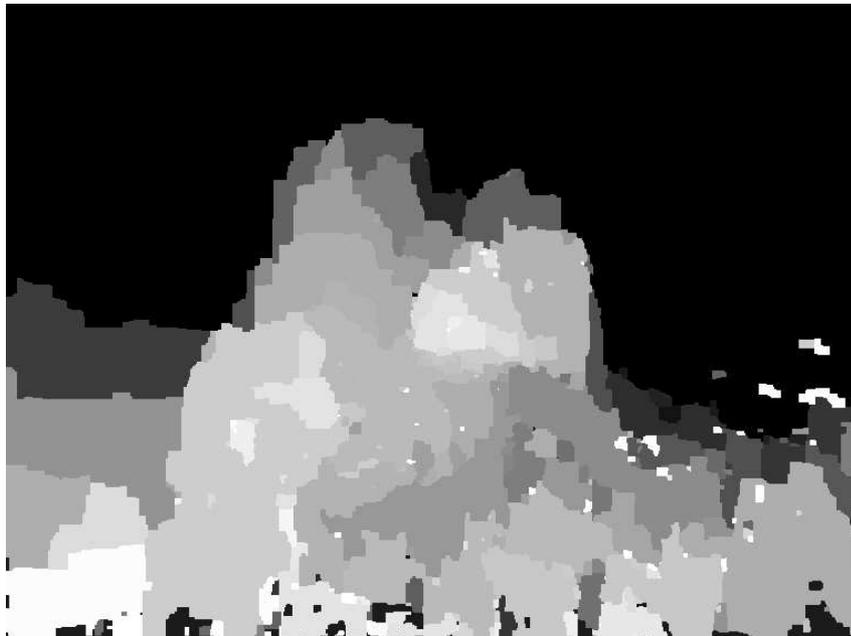
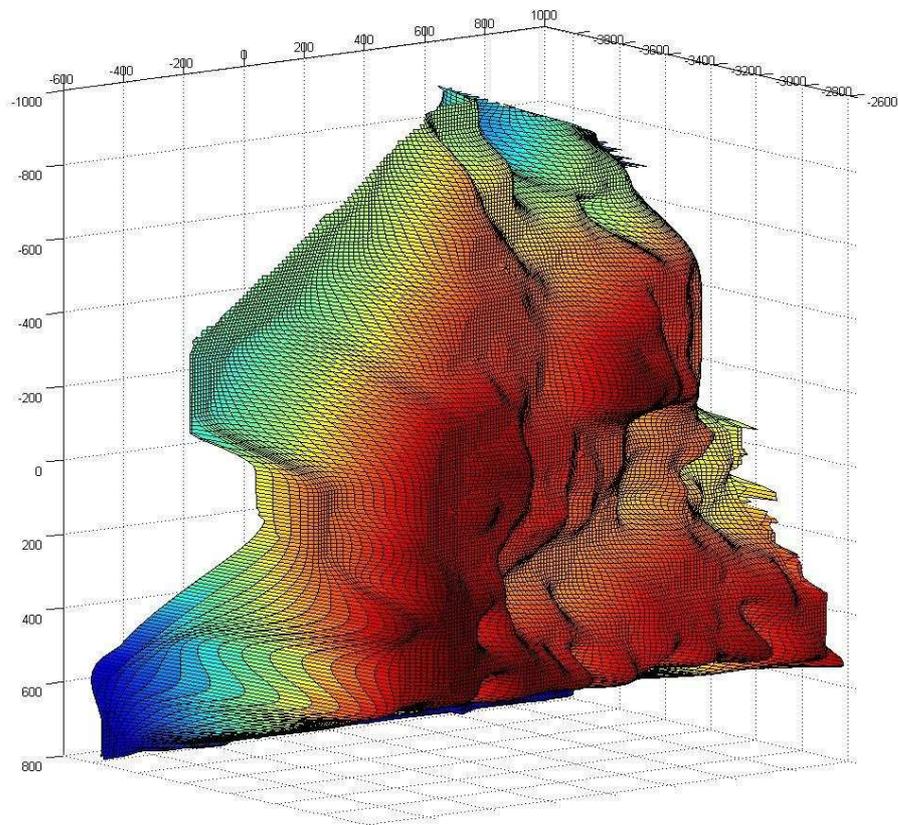


FIG. 3.30 – Carte de disparité dense avec la méthode de Graph-Cut



(a) Carte de profondeur



(b) Structure 3D avec plaquage de texture

3.2.2 Evaluation de la précision des mesures sur la structure 3D

Dans notre cas, l'intérêt principal de calculer des reconstructions 3D d'objets sous-marins est de pouvoir faire ensuite dessus des mesures métriques les plus précises possibles, en se basant exclusivement sur les images. C'est pourquoi nous avons utilisé en premier exemple des objets de formes géométriques simples à mesurer, pour évaluer la méthode de reconstruction. Les mesures effectuées sont indiquées directement sur la reconstruction 3D dans la figure 3.32 et reportées dans le tableau 3.33. Les mesures réelles effectuées directement sur l'objet apparaissent entre parenthèses sur la figure. Plusieurs mesures sont proches des mesures réelles à quelques millimètres près, mais l'erreur de mesure atteint tout de même 15mm sur une mesure réelle de 240mm (ce qui fait une erreur de 6,25%). Ceci est dû à des déformations locales de la reconstruction 3D suite à de mauvais appariements qui faussent la mesure. Pour améliorer la précision des mesures, il faut donc améliorer la reconstruction 3D en prenant en compte davantage de vues de l'objet pour éviter les mauvais appariements.

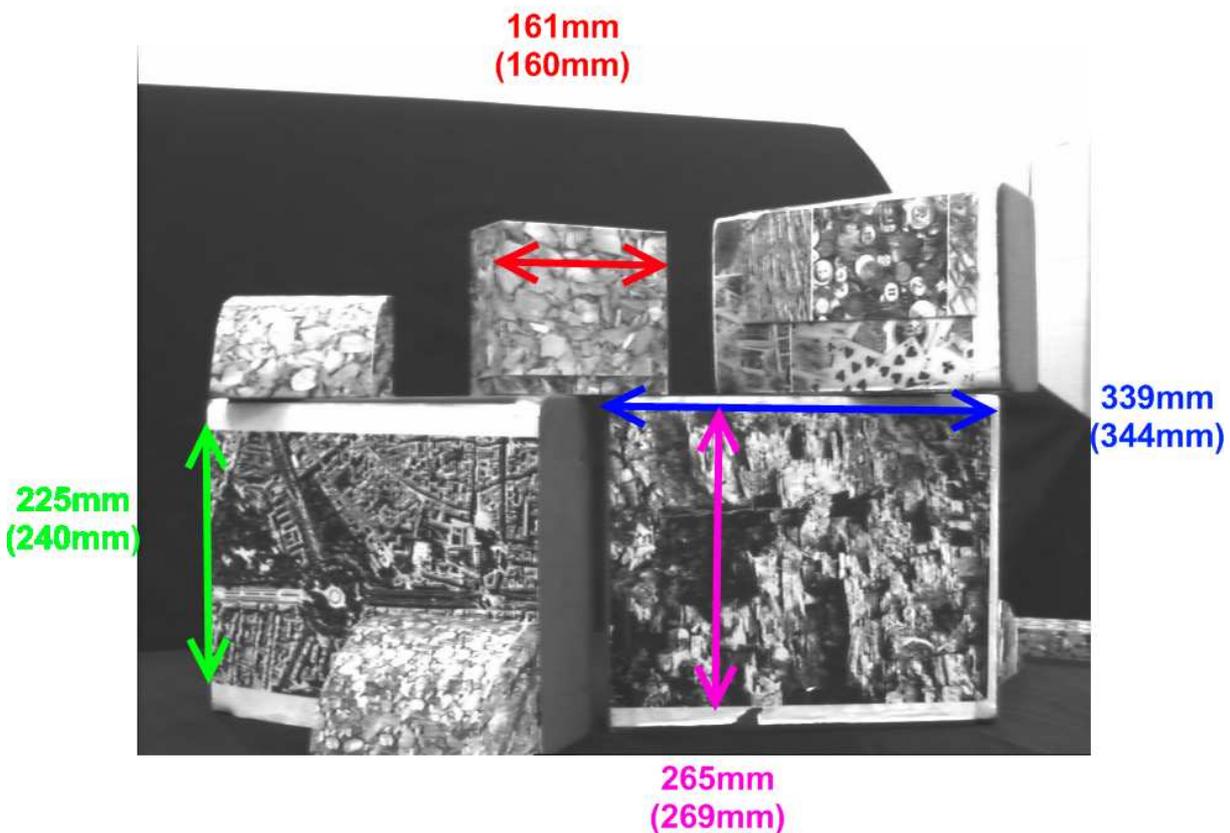


FIG. 3.32 – Mesures métriques sur une structure 3D

		1	2	3	4
Point 1 (mm)	X	-326,8	-81,8	18,2	118,2
	Y	1,2	-138,8	-8,8	-13,8
	Z	1001,0	936,7	957,4	966,2
Point 2 (mm)	X	-326,8	78,2	353,2	118,2
	Y	226,2	-138,8	-8,8	251,2
	Z	1005,0	956,3	1011,0	978,9
D_{12} mesurée (mm)		225	161	339	265
D_{12} réelle (mm)		240	160	344	269

FIG. 3.33 – Fiabilité des mesures métriques faites sur une reconstruction 3D obtenue à partir d'une paire d'images

3.3 Conclusion

La méthode de reconstruction exposée dans ce chapitre ne représente pas une nouveauté en soi, mais elle s'articule autour d'algorithmes connus et performants, adaptés au type d'objets et de reconstruction que l'on désire obtenir dans le cadre de ce travail. Cette méthode s'appuie sur une estimation des paramètres des caméras obtenus par calibration à l'aide d'une mire, et sur l'estimation de chaque déplacement des caméras correspondant à la géométrie de la tête stéréo.

Les différents exemples de reconstructions présentés ici, illustrent bien la robustesse et la précision de la méthode. Ainsi on a utilisé un cas de reconstruction avec plusieurs objets de formes facilement identifiables et mesurables, qui nous a permis d'estimer la précision de la méthode. Un second cas nous a permis de tester la méthode sur un objet de taille moyenne, se rapprochant des cheminées hydrothermales par sa forme et sa texture. Le dernier cas a validé la méthode sur des images prises en conditions réelles d'expérimentation en mer avec un système stéréo attaché en bout de bras d'un robot sous-marin.

Les résultats obtenus sur seulement deux images sont prometteurs. En étendant la méthode à l'ensemble des images de la trajectoire, et en fusionnant les données on peut s'attendre à améliorer encore davantage la précision de la reconstruction 3D.

Conclusion et perspectives

L'objectif de cette thèse est de proposer une méthodologie complète de reconstruction 3D d'objets sous-marins naturels répondant à un besoin scientifique de mesures quantitatives *in situ*, qui puisse être utilisée en conditions opérationnelles. La première difficulté a été de prendre en compte les différents problèmes liés au milieu sous-marin profond ainsi que les moyens techniques disponibles pour effectuer la capture des images à l'aide du robot. Ainsi il a fallu trouver une méthode permettant l'acquisition automatique d'images, s'adaptant à tout type d'objet sous-marin naturel de faible échelle (environ $1m^3$), tout en maîtrisant la connaissance de la position et de l'orientation de la caméra afin de faciliter la reconstruction 3D. Une méthode complète s'appuyant sur une technique d'asservissement visuel utilisant un système stéréo pour la capture d'images à intervalles réguliers a donc été développée. Grâce à cette approche, chaque position de la caméra est connue, puisque l'intervalle entre deux images successives correspond à la géométrie de la tête stéréo.

Après avoir présenté dans ce manuscrit les différentes options s'offrant à nous pour réaliser l'asservissement visuel et définir la structure 3D à partir des images acquises, la méthodologie a été détaillée en deux étapes : une partie consacrée à la méthode d'acquisition des images, et une autre consacrée à l'exploitation de ces images afin d'en obtenir une reconstruction 3D. Nous avons montré analytiquement que l'application d'une vitesse constante de translation et de rotation à une caméra, lui fait décrire une trajectoire qui reste toujours sur la surface d'un cylindre.

La mise en oeuvre de la génération de trajectoires par asservissement visuel entre deux caméras, placées en bout de robot manipulateur, a d'abord été faite en simulation en y intégrant la cinématique du bras et les paramètres des caméras. Cette première étape vers un système opérationnel a permis de valider notre approche en vue de préparer les expérimentations. Le dimensionnement de la tête et la programmation du bras en fonction de l'espace de travail disponible ont été faits à partir des simulations. Celles-ci ont également représenté un outil précieux par la suite pour l'analyse des résultats expérimentaux. Ainsi on a pu recréer et observer par simulation l'influence des erreurs d'appariement, la dérive des points suivis dans l'image, ou encore l'influence du bruit sur la commande.

Un aspect particulièrement important de ce travail était d'obtenir des résultats expérimentaux en milieu sous-marin, ce qui a mobilisé d'importants moyens matériels et humains comme plusieurs bras manipulateurs, un robot sous-marin embarqué sur un navire océanographique, des caméras sous-marines, et des bassins d'expérimentations, avec l'aide de tout le personnel compétent. Pour ce faire, des premières expérimentations ont été menées dans l'air, pour tester la faisabilité de plusieurs déplacements de la tête stéréo dans l'espace de travail du bras manipulateur, par asservissement visuel face à plusieurs objets plans. La

méthode de suivi ESM [BM04b] utilisée permettait de suivre des plans. L'asservissement a été testé avec une commande 2D dans l'image dite « classique », et une commande invariante aux paramètres intrinsèques des caméras [Mal01a]. Les résultats obtenus étant conformes à ceux obtenus précédemment en simulation, les mêmes expérimentations ont été conduites sur un robot représentant la réplique du bras manipulateur sous-marin, avec cette fois des cibles non-planaires, de formes et de tailles différentes. D'une manière générale, les résultats ont montré une grande sensibilité de l'asservissement visuel aux erreurs de suivi des points et d'appariement. Le suivi dans l'image se faisait alors par l'algorithme KLT [LK81], permettant un suivi indépendant des points extraits et appariés dans les images stéréo. De plus, la commande 2D est sensible aux erreurs de calibration, alors que la commande invariante est plus sensible au bruit et à l'estimation des profondeurs. Les résultats ont également montré une meilleure précision en positionnant le bras par « programmes robots » (asservissement électrohydraulique de position grâce aux informations provenant de ses capteurs de positions articulaires), mais la méthode est moins flexible, et elle est entièrement dépendante de la cinématique et du réglage des asservissements du bras. Des expérimentations ont également été menées en bassin, puis à l'occasion d'une campagne en mer sur la dorsale océanique au large des Açores. Ainsi, des images sous-marines ont pu être acquises en milieu profond par le système stéréo baptisée IRIS, le long d'une trajectoire à intervalles réguliers, afin d'être exploitées pour tester la méthode de reconstruction 3D.

Plusieurs approches de reconstruction 3D ont été étudiées et parfois testées comme le *space carving* [KS00]. Néanmoins celle retenue, car mieux adaptée à notre problématique, en prenant en compte le type d'objet à reconstruire, les problèmes d'éclairage, le bruit dans l'image ou l'encombrement matériel, est celle à partir de la géométrie épipolaire. Cette méthode fait partie de la stratégie mise en place pour arriver à la structure 3D de l'objet en s'appuyant sur les données obtenues avec l'acquisition par asservissement visuel, mais elle ne constitue pas une nouveauté. Dès lors nous nous sommes appuyés sur des algorithmes connus et performants (SIFT, Bundle adjustment, Graph-Cut, ...), que nous avons sélectionnés pour leurs résultats sur nos images de scènes naturelles sous-marines, puis intégrés dans un même processus de reconstruction. La méthode prend en compte une estimation des paramètres des caméras obtenus après calibration à l'aide d'une mire, ainsi que l'estimation de chaque déplacement des caméras correspondant à la géométrie de la tête stéréo. Des résultats sur plusieurs exemples très différents, que ce soit par rapport au type d'objet ou au milieu environnant, ont été obtenus à partir d'une seule paire d'image stéréo. L'ampleur du travail effectué pour définir une stratégie de reconstruction 3D, simuler nos algorithmes, puis expérimenter le système en respectant de nombreuses étapes avant d'atteindre les conditions opérationnelles, n'a pas permis par manque de temps d'aller jusqu'à une reconstruction 3D exploitant toutes les images acquises le long de la trajectoire. Il aurait encore fallu pouvoir fusionner les données obtenues par l'appariement dense entre les paires d'images stéréo (seule l'estimation de la structure 3D à partir des points SIFT extraits dans sept images le long d'une même trajectoire a pu être validée). Cependant les résultats obtenus en terme de rendu et de précision sur seulement deux images, montrent les capacités de la méthode sur des images sous-marines réelles à fournir des mesures fiables sur le modèle 3D.

La continuité naturelle de ce travail serait évidemment d'arriver à un résultat exploitant

toutes les images disponibles, en intégrant la notion de fusion de données à notre travail. Les positions articulaires lues par les résolveurs à chaque articulation du bras pourraient être exploitées pour reconstruire la position des caméras. De plus, la plupart des expérimentations d'asservissement visuel ont été menées dans l'air (principalement pour des raisons de mise en œuvre moins contraignante), il serait donc également intéressant de poursuivre les expérimentations d'asservissement visuel en milieu sous-marin, pour perfectionner celui-ci face aux perturbations que l'on ne rencontre pas dans l'air (particules en suspension, éclairage non-uniforme, objets en mouvement, modification des paramètres intrinsèques avec la profondeur,...). En effet, la dérive des points due à ses perturbations est une cause importante d'erreur au niveau de l'asservissement visuel. Enfin, il serait utile de pouvoir réellement comparer cette méthode de reconstruction 3D, en terme de résultat et d'efficacité, par rapport aux autres méthodes existantes qui ont été écartées dès le début de nos recherches.

Annexe A

Logiciel pour l'acquisition d'images stéréoscopiques et pour l'asservissement visuel

Cette annexe décrit l'utilisation d'un logiciel d'asservissement visuel, développé pour une utilisation avec un système de stéréovision et un système robotique asservi en vitesse, pour l'acquisition d'images permettant la reconstruction 3D d'objets sous-marins.

A.1 Fonctionnalités

La fonction principale de ce logiciel est de générer une commande à partir des images extraites des caméras, afin de réaliser un asservissement visuel entre les deux caméras du système stéréo, l'une représentant la position à partir de laquelle est capturée l'image de référence, l'autre étant la caméra à asservir sur cette même image. L'espace et l'angle entre les deux caméras, qui définissent la géométrie de la tête stéréo, contraignent les caméras placées en bout de bras d'un robot manipulateur à suivre une trajectoire connue avec la répétition de l'asservissement. Cette technique de génération de trajectoires permet de connaître les positions des caméras après chaque asservissement, facilitant ainsi le travail des algorithmes de reconstruction 3D utilisés sur les images sous-marines. Une commande en vitesse dans le repère pince est envoyée par une liaison série RS232 au PC du bras manipulateur. Cette commande est calculée à partir des informations extraites des images des caméras, et d'un certain nombre de paramètres de réglages des caméras, à renseigner dans l'interface du logiciel. Le logiciel permet également l'archivage d'images et de séquences vidéos des caméras stéréo.

A.2 Interface graphique

L'interface graphique est divisée par rapport au nombre d'étapes nécessaires afin de pouvoir générer la commande (cf. figure A.2). Il existe donc quatre étapes qui sont :

- Configuration des caméras et affichage ;
- Spécification de la tête stéréo ;

- Suivi dans l'image ;
- Envoi de la commande.

L'interface permet de visualiser les images des deux caméras en même temps. Complètement à droite de l'interface un certain nombre de boutons sont réunis permettant le test des caméras, et l'affichage des images, la sauvegarde des images ou des séquences vidéos ainsi que les chemins de sauvegarde sur le disque dur, et le bouton de fermeture de l'application.

A.2.1 Configuration des caméras et affichage

Cette partie de l'interface (cf. figure A.2.1) permet de choisir les images à afficher, et de surveiller l'espace libre du disque dur pour la sauvegarde des images, ainsi que le délai restant pour la sauvegarde des séquences vidéos en fonction de la cadence d'écriture choisie.

A.2.2 Spécifications de la tête stéréo

Le bouton configuration de la fenêtre principale (cf. figure A.3.a) permet d'ouvrir une autre fenêtre pour saisir les différents paramètres des caméras (cf. figure A.3.b). La partie supérieure correspond aux paramètres intrinsèques des caméras qui correspondent à :

- la focale f de la caméra ;
- k_u et k_v : les facteurs d'agrandissement de l'image ;
- u_0 et v_0 : les coordonnées de la projection du centre optique de la caméra sur le plan image.

La partie inférieure permet de saisir la géométrie de la tête stéréo (écartement et angle entre les caméras), la position de la caméra gauche par rapport à la pince, ainsi que la position de saisie de la tête stéréo par la pince (au-dessus ou derrière les caméras).

Tous ces paramètres permettent de calculer par triangulation les profondeurs des points extraits et appariés entre les images des deux caméras. La position de la pince par rapport à la caméra gauche (caméra asservie) permet d'exprimer la commande dans le repère pince au lieu du repère caméra.

A.2.3 Suivi dans l'image

Une fois les caméras configurées, et la visualisation des images lancée, on réalise l'extraction et l'appariement des points entre les deux images par le bouton « TRACKING ON ». Les points étant ainsi mis en correspondance, seuls les points dans l'image de la caméra gauche sont suivis en temps réel. Les points et l'image de la caméra droite restent figés afin de garder les points de référence affichés à l'écran. Dès lors le bouton change de label et devient « TRACKING OFF » pour arrêter le suivi. Un certain nombre de paramètres permettent de définir la taille et la position de la fenêtre où l'on souhaite extraire et suivre les points (cf. figure A.4.a). On peut choisir également de centrer la fenêtre dans l'image, et permettre ou non le déplacement de cette fenêtre suivant le déplacement des points suivis dans l'image. Le dernier paramètre est le nombre de points KLT extraits dans l'image gauche, candidats pour être appariés avec un point dans l'image de la caméra

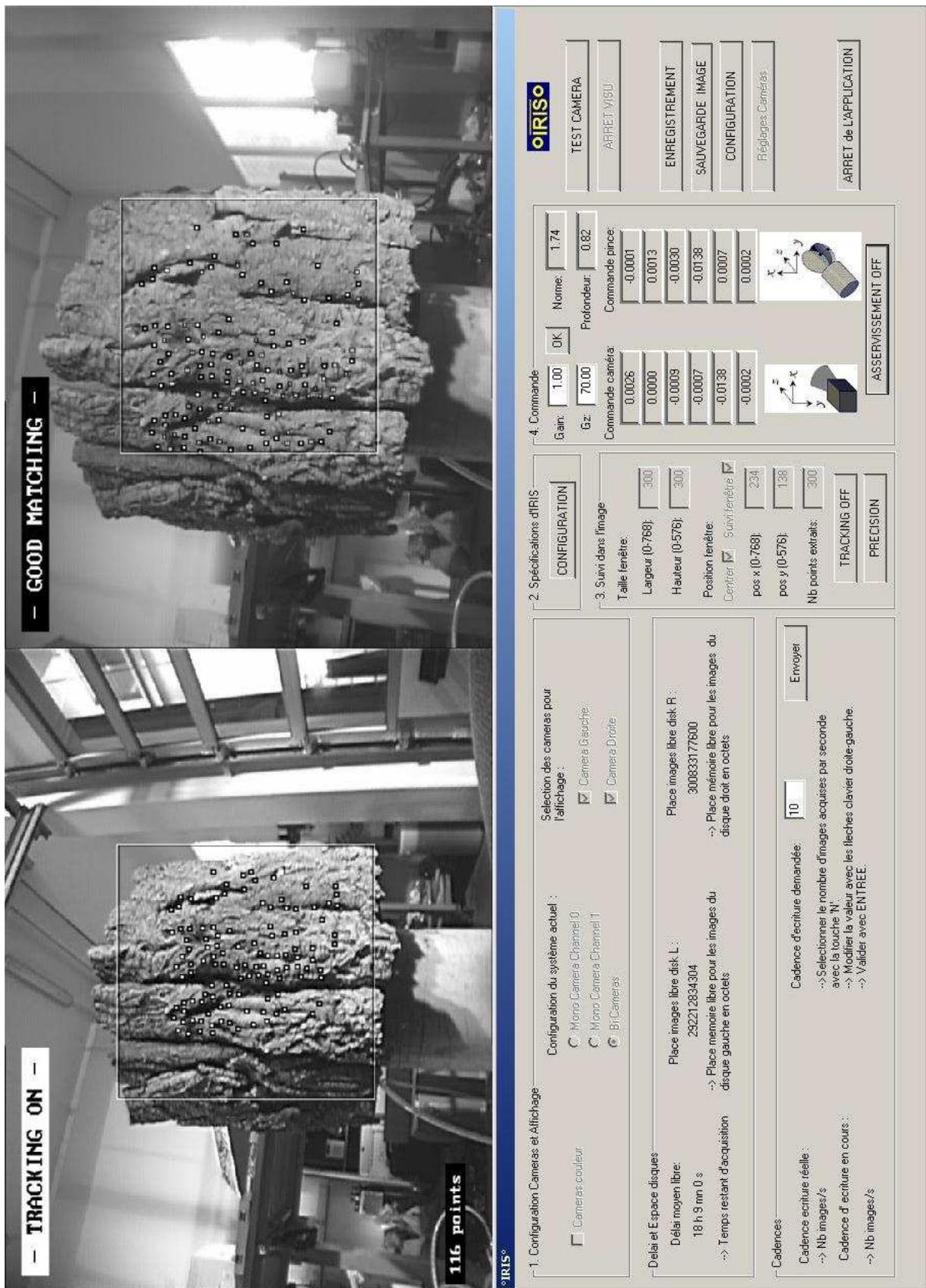


FIG. A.1 – Interface graphique du logiciel pour l'asservissement visuel

1. Configuration Caméras et Affichage		
<input type="checkbox"/> Caméras couleur	Configuration du système actuel :	Selection des caméras pour l'affichage :
	<input type="radio"/> Mono Camera Channel 0	<input checked="" type="checkbox"/> Camera Gauche
	<input type="radio"/> Mono Camera Channel 1	<input checked="" type="checkbox"/> Camera Droite
	<input checked="" type="radio"/> Bi Caméras	
Délai et Espace disques		
Délai moyen libre:	Place images libre disk L :	Place images libre disk R :
18 h 9 mn 0 s	292212834304	300833177600
--> Temps restant d'acquisition	--> Place memoire libre pour les images du disque gauche en octets	--> Place mémoire libre pour les images du disque droit en octets
Cadences		
Cadence ecriture réelle :	Cadence d'ecriture demandée:	<input type="text" value="10"/> <input type="button" value="Envoyer"/>
--> Nb images/s	--> Selectionner le nombre d'images acquises par seconde avec la touche 'N'.	
Cadence d' ecriture en cours :	--> Modifier la valeur avec les fleches clavier droite-gauche.	
--> Nb images/s	--> Valider avec ENTREE.	

FIG. A.2 – Configuration de l’affichage des caméras et de l’enregistrement des images

droite. Il ne s’agit donc pas forcément du nombre de points suivis au départ de l’asservissement, mais d’un nombre maximum de points pouvant être mis en correspondance. Plus ce nombre est grand, plus le critère de sélection de points robustes dans l’image diminue, et plus le traitement est long, mais plus on a de chance de conserver un nombre suffisant de points jusqu’à la fin de l’asservissement visuel. La réussite du processus d’appariement est indiqué sur l’écran par « GOOD MATCHING », et en cas d’échec par « BAD MATCHING ». De même si le suivi de points est actif on aura l’affichage de « TRACKING ON », et si celui-ci est arrêté on aura donc « TRACKING OFF ». Le nombre de points suivis est affiché en temps réel en bas à gauche de l’image de la caméra gauche (cf. figure A.4.b). Un dernier bouton nommé « PRECISION » permet d’éliminer des points qui auraient dérivés dans l’image durant l’asservissement visuel. Cette fonction compare les points suivis aux points de l’image de référence par un produit de corrélation. Il est donc conseillé d’utiliser cette fonction uniquement en fin d’asservissement afin d’éliminer d’éventuelles erreurs de suivi, et donc d’améliorer la précision de l’asservissement.

A.2.4 Envoi de la commande au robot

Une fois les points appariés et suivis dans l’image gauche, la valeur de la commande apparaît dans la partie 4 de l’interface graphique. Cette commande représente les vitesses cartésiennes à appliquer au repère de la caméra gauche pour rejoindre la position de référence représentée par la caméra droite au moment de l’acquisition de l’image de référence. Un « Gain » permet d’ajuster la vitesse de convergence de l’asservissement. Le risque en le mettant trop fort est de perdre tous les points suivis. Un deuxième gain « Gz » permet



(a)

Estimation paramètres intrinsèques

Paramètre	Caméra droite	Caméra gauche
focale	7.45 mm	5.55 mm
Ku	100.00 px/mm	100.00 px/mm
Kv	100.00 px/mm	100.00 px/mm
Uo	388.98 px	373.23 px
Vo	297.46 px	297.96 px

IRISO

DROITE GAUCHE

Géométrie IRIS

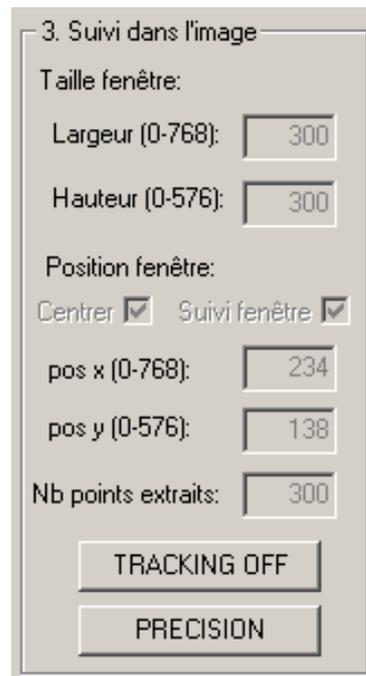
Ecartement des caméras	30.00 cm
Angle de la caméra mobile	15.00 °
Ecartement caméra gauche - pince	15.00 cm
Mode de saisie d'IRIS par la pince	<input type="radio"/> Dessus <input checked="" type="radio"/> Derrière

OK

(b)

FIG. A.3 – Paramètres de la tête stéréo

de moduler l'estimation de la profondeur des points à un facteur près (si les caméras sont bien calibrées il vaut mieux le laisser à 1).



(a) Configuration de la fenêtre de suivi et du nombre de points extraits dans l'image



(b) Visualisation du suivi de points dans l'image de la caméra asservie

FIG. A.4 – Configuration et visualisation du suivi de points

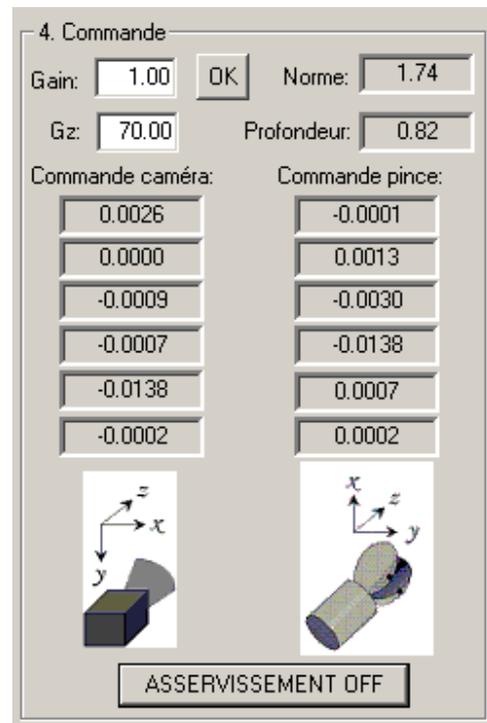


FIG. A.5 – Visualisation de la commande dans le repère caméra et dans le repère pince avant d'être appliquée sur le bras manipulateur

Glossaire

- Ajustement de faisceaux** : Etant donné un ensemble d'images illustrant un certain nombre de points 3D à partir de différents points de vue, l'ajustement de faisceaux est une méthode permettant d'affiner conjointement l'ensemble des coordonnées des points 3D et les paramètres des caméras ayant servi à l'acquisition des images, en fonction d'un critère d'optimisation impliquant la projection dans les images correspondantes de tous les points.
- Appariement** : mise en correspondance de formes représentant les projections de la même entité de la scène, selon un ensemble prédéfini de règles ou de critères.
- Asservissement (automatique)** : un asservissement en automatique est un algorithme dont l'objet principal est de stabiliser et d'améliorer la réaction d'un système par rapport à sa consigne. Le principe général est de comparer la consigne et l'état du système de manière à le corriger.
- Asservissement visuel** : Le principe de l'asservissement visuel consiste à prendre en compte des informations visuelles issues d'une ou plusieurs caméras dans la boucle de commande d'un robot afin d'en contrôler le mouvement. Les lois de commande consistent alors à contrôler le mouvement de la caméra afin que les mesures dans l'image atteignent une valeur désirée, voire suivent une trajectoire.
- Automatique** : L'automatique fait partie des sciences de l'ingénieur. Cette discipline traite de la modélisation, de l'analyse, de la commande et, de la régulation des systèmes dynamiques. Elle a pour fondements théoriques les mathématiques, la théorie du signal et l'informatique théorique. L'automatique permet l'automatisation de tâches par des machines fonctionnant sans intervention humaine. On parle alors de système asservi ou régulé.
- Bruit** : terme générique regroupant des parasites de diverses origines (surtout électroniques), affectant la restitution d'un signal audio ou vidéo.
- Calibration de caméra** : L'utilisation d'une caméra pour mesurer les distances métriques et angulaires qui la séparent des formes polyédriques 3D ou 2D, contenues dans un espace tridimensionnel, nécessite d'avoir une connaissance des paramètres internes de la caméra : les paramètres intrinsèques. Ainsi la calibration (anglicisme de calibrage) d'une caméra permet de caractériser les paramètres intrinsèques et extrinsèques. Pour ce faire, on utilise un élément connu en dimension pour modéliser l'ensemble de la caméra (objectif, échantillonnage, bruits...).
- CCD** : Charge Coupled Device, dispositif à transfert de charge. Ce capteur d'images transforme l'énergie lumineuse en énergie électrique constituant une mosaïque

mesurée en fraction de pouce. Les capteurs CCD existent en format 2/3", 1/2", 1/3" et 1/4".

Corrélation : mesure permettant d'évaluer la ressemblance entre un pixel de l'image gauche (et son voisinage) et un pixel de l'image droite (et son voisinage).

Détecteur de points d'intérêts : la détection de points d'intérêts est, au même titre que la détection de contours, une étape préliminaire à de nombreux processus de vision par ordinateur. Les méthodes de détection de point d'intérêts peuvent être classées grossièrement suivant trois catégories : les approches contours, intensité, ou à base de modèles.

Diagramme de Voronoï : Un diagramme de Voronoï est généré à partir d'un ensemble E de points, appelés sites ou germes, appartenant au plan. Chaque point de E est à l'intérieur d'un polygone convexe qui délimite une surface formée des points du plan qui sont plus proches de ce site que des autres sites. Chaque polygone obtenu est appelé polygone de Voronoï. Les sommets de Voronoï sont les différents sommets de chaque polygone et les arêtes de Voronoï sont donc constituées des points à égale distance de deux sites.

Disparité : vecteur de déplacement entre le pixel de l'image gauche et son correspondant dans l'image de droite.

Focale : distance séparant le centre optique d'un objectif (les rayons passant par son centre ne sont pas déviés), du point de convergence (foyer) des rayons parallèles incidents (soleil ou sources lointaines). La distance focale s'exprime en mm ou en pixel.

Fonction de tâche : outil mathématique permettant d'établir l'influence mutuelle de plusieurs tâches, quel que soit le capteur ou la stratégie de commande associés. l'intégration de l'asservissement visuel dans l'approche générale de la fonction de tâche permet de résoudre de manière efficace et élégante les problèmes de redondance rencontrés lorsqu'une tâche visuelle ne contraint pas l'ensemble des degrés de liberté de la caméra. Il est alors possible d'exécuter, parallèlement à la tâche visuelle, un ensemble de tâches secondaires comme des suivis de trajectoires par exemple.

Géométrie épipolaire : décrit la relation entre deux images d'une même scène, qui est caractérisée par une matrice (3×3) singulière. Cette matrice est connue sous le nom de matrice essentielle si les paramètres intrinsèques des images sont connus, de matrice fondamentale s'ils ne le sont pas. Elle contient toutes les informations géométriques des deux images. Son estimation est très importante pour beaucoup d'applications comme la modélisation de scènes et le déplacement d'un robot mobile autonome.

KLT : l'algorithme de Kanade-Lucas-Tomasi (KLT) est une méthode largement utilisée en vision par ordinateur aussi bien pour l'alignement ou le recalage d'images, que pour le suivi de régions d'intérêt dans une séquence d'images.

Lampe HMI : Hydrargyrum medium-arc iodide, lampe aux halogénures métalliques ou lampe à décharge à haute pression, synonyme de lampe à arc, signifie que la

lumière est créée par un arc électrique dans un gaz.

Loi de commande : la loi de commande permet au régulateur de calculer la meilleure correction à effectuer pour ramener le plus efficacement possible la mesure à la valeur de consigne.

Lumière structurée : simplifier la mise en correspondance en remplaçant une caméra par un projecteur de motifs.

Maillage : arrangement géométrique des pixels dans l'image. Il existe 3 types de tessellation du plan par des figures géométriques : maillage carré, hexagonal, ou triangulaire.

Matrice d'interaction : matrice reliant une variation de l'image au mouvement de caméra qui induit cette variation. Il est ainsi possible de déduire des modifications observées dans l'image, le mouvement de caméra qui en est à l'origine. À l'inverse, on peut également en déduire le déplacement que doit faire le robot pour converger vers l'image apprise correspondant à la position désirée du robot.

Paramètres d'une caméra : les paramètres de la caméra sont classés en deux catégories : les paramètres intrinsèques qui traduisent les propriétés de l'optique géométrique et du système d'acquisition de la caméra, et les paramètres extrinsèques qui traduisent la position et l'orientation de la caméra par rapport à un référentiel externe.

Pixel : picture element, unité de base de l'image correspondant à un pas de discrétisation.

Plaquage (ou mappage) de texture : technique visant à dessiner un objet de manière à ce que les surfaces apparaissant sur cet objet soient revêtues d'une image. Le placage de texture est très largement utilisé en synthèse d'images pour améliorer la qualité des vues obtenues.

Point d'intérêt : Les points d'intérêts, dans une image, correspondent à des doubles discontinuités de la fonction d'intensités. Celles-ci peuvent être provoquées, comme pour les contours, par des discontinuités de la fonction de réflectance ou des discontinuités de profondeur. Ce sont par exemple : les coins, les jonctions en T ou les points de fortes variations de texture.

Primitive : du point de vue du traitement numérique d'images, il s'agit de représentations des formes, manipulables par un ordinateur, afin d'être capable de les identifier dans une image numérique. Ces primitives peuvent être d'ordre métrique, analytique, structurel ou symbolique.

Régulateur : mécanisme automatique qui élabore un signal de commande U en fonction de l'écart de réglage $M - C$ selon un algorithme donné $f : U = f(M - C)$.

Résolveur : désigne dans l'électrotechnique un transducteur électromagnétique pour le changement d'angle d'un rotor en une valeur électrique.

Rectification d'images : consiste à recalculer, pour deux images en position générale, deux nouvelles images telles que la géométrie épipolaire de ces deux images soit *simple* ; c'est à dire que les droites épipolaires sont horizontales, ce qui implique que les deux nouveaux épipôles soient à l'infini.

ROV : *Remote Operated Vehicle*, véhicule sous-marin téléopéré capable d'effectuer de l'inspection vidéo ou instrumenté (caméras, moyens de prélèvement, capteurs

scientifiques et de navigation), de la recherche et des opérations de mesures. Le ROV Victor 6000 est un Véhicule d'Intervention par Câble TéléOpéré pour la Recherche capable d'intervenir jusqu'à 6000 m de profondeur.

SIFT : le descripteur d'image SIFT (Scale Invariant Feature Transform) a été introduit dans [Low04]. Il a été conçu pour être invariant à la fois aux changements d'échelles et aux rotations. De plus il est robuste aux transformations affines, au bruit et aux variations d'illuminations.

Stéréophotogrammétrie : consiste à mesurer la taille des objets et la position des objet dans l'espace à partir d'un ensemble d'images stéréoscopique.

Texture : Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont par exemple un motif basique qui se répète, ou des caractéristiques fréquentielles. Une texture est composée de texels, l'équivalent des pixels.

Traitement d'images : discipline des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information

Trajectoire : ensemble de tous les points successivement occupés par un objet en mouvement.

Triangulation : calcul point 3D / Maillage triangulation de Delaunay maillage tessellation

Triangulation de Delaunay : La triangulation de Delaunay d'un ensemble de n points est l'unique triangulation telle qu'un cercle passant par les trois points d'un triangle ne contienne aucun autre point, c'est le dual du diagramme de Voronoï.

Vision 3D : branche commune entre les domaines très liés de la photogrammétrie et de la vision par ordinateur ; elle étudie comment obtenir des informations tridimensionnelles, à partir d'images bi-dimensionnelles. Le but recherché peut être l'obtention de mesures sur des objets, la création de modèles photoréalistes, la localisation de la caméra, ou la mesure d'un déplacement.

Vision active : la vision active peut être définie comme un contrôle actif de tous les paramètres du capteur de vision afin de réaliser une tâche donnée.

Vision par ordinateur : aussi appelée vision artificielle, vision numérique ou plus récemment vision cognitive, représente une branche de l'intelligence artificielle, correspondant à l'étude des différentes phases principales issues du processus de traitement des images (formation, analyse et interprétation des images).

Vision stéréoscopique : la vision stéréoscopique peut être définie comme la possibilité de percevoir les distances relatives des objets sur la base de l'information visuelle provenant de la disparité binoculaire.

Bibliographie

- [ABH⁺05] A.G. ALLAIS, V. BRANDOU, U. HOGE, M. BERGMANN, J.P. LEVEQUE, P. LEON, J.F. CADIOU, J. SARRAZIN et P.M. SARRADIN : Design of optical instrumentation for 3d and temporal deep-sea observation. *In Proc. of the 1st international conference of optical complex systems*. Marseille, France, 2005.
- [BAC06] Boulbaba BEN AMOR, Mohsen ARDEBILIAN et Liming CHEN : Enhancing 3D Face Recognition By Mimics Segmentation. *In the 6th International Conference on Intelligent Systems Design and Applications (ISDA'06)*, octobre 2006.
- [Bar03] A. BARTOLI : *Reconstruction et alignement en vision 3D : points, droites, plans et caméras*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Septembre 2003.
- [BB97] Edmond BOYER et M.-O. BERGER : 3d surface reconstruction using occluding contours. *International Journal of Computer Vision*, 22(3):219–233, 1997.
- [BC01] Jasmine BANKS et Peter CORKE : Quantitative evaluation of matching methods and validity measures for stereo vision. *The International Journal of Robotics Research*, 20(7):512–532, July 2001.
- [Bea78] P. BEAUDET : Rotationally invariant image operators. *In Proc. 4th Int. Joint Conf. Patt. Recog.*, pages 579–583. Kyoto, Japan, 1978.
- [Bir07] S. BIRCHFIELD : Kanade-lucas-tomasi (klt) feature tracker. *In <http://www.ces.clemson.edu/stb/klt>*, (accessed 2007).
- [BM01] Simon BAKER et Iain MATTHEWS : Equivalence and efficiency of image alignment algorithms. *In Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1090 – 1097, December 2001.
- [BM04a] Simon BAKER et Iain MATTHEWS : Lucas-kanade 20 years on : A unifying framework. *International Journal of Computer Vision*, 56(3):221 – 255, March 2004.
- [BM04b] S. BENHIMANE et E. MALIS : Real-time image-based tracking of planes using efficient second-order minimization. *In IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 943–948. Sendai, Japan, October 2004.

- [BMR⁺06] V. BRANDOU, E. MALIS, P. RIVES, A.G. ALLAIS et M. PERRIER : Active stereo-vision using invariant visual servoing. *In IEEE International Conference on Intelligent Robots and Systems*. Beijing, China, 2006.
- [Bou99] J.Y. BOUGUET : Pyramidal implementation of the lucas-kanade feature tracker. *In OpenCV Documentation, Microprocessor Research Labs, Intel Corporation*, 1999.
- [BP98] J.Y. BOUGUET et P. PERONA : Camera calibration from points and lines in dual-space geometry. Rapport technique, California Institute of Technology, 1998.
- [Cha90] F. CHAUMETTE : *La relation vision-commande : théorie et application à des tâches robotiques*. Thèse de doctorat, Université de Rennes I, IRISA, Juillet 1990.
- [CKS94] W. Z. CHEN, U. A. KORDE et S. B. SKAAR : Position control experiments using vision. *In Int. Journal of Robotics Research*, volume 13(3) :199–208, Juin 1994.
- [CRE91] F. CHAUMETTE, P. RIVES et B. ESPIAU : Positioning a robot with respect to an object tracking it and estimating its velocity by visual servoing. *In IEEE International Conference on Robotics and Automation*, pages 2248–2253. Sacramento, California, April 1991.
- [CS00] Sung-Il CHIEN et Si-Hun SUNG : Adaptive window method with sizing vectors for reliable correlation-based target tracking. *Pattern Recognition*, 33(2):237–249, 2000.
- [ECFP92] B. ESPIAU, CHAUMETTE, F. et Rives P. : A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.
- [EGAG96] Grosso E., Metta G., Oddera A. et Sandini G. : Robust visual servoing in 3-d reaching tasks. *In IEEE Transactions on Robotics and Automation*, volume 12 (5), pages 732–742, 1996.
- [F94] Wolfgang FÖRSTNER : A framework for low level feature extraction. *In ECCV '94 : Proceedings of the third European conference on Computer Vision (Vol. II)*, pages 383–394, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
- [Fal94] L. FALKENHAGEN : Depth estimation from stereoscopic image pairs assuming piecewise continuous surfaces. *In Proc. of European Workshop on combined Real and Synthetic Image Processing for Broadcast and Video Production*, Hamburg, 1994.
- [FB81] M. A. FISCHLER et R. C. BOLLES : Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *In Comm. of the ACM*, volume 24, pages 381–395, 1981.
- [FG87] W. FÖRSTNER et E. GÜLCH : A fast operator for detection and precise location of distinct points, corners and centres of circular features. *In Intercommission Conference on Fast Proc. of Photogrammetric Data*, pages 281–305, 1987.

-
- [FLM92] Olivier D. FAUGERAS, Quang-Tuan LUONG et Stephen J. MAYBANK : Camera self-calibration : Theory and experiments. *In ECCV '92 : Proceedings of the Second European Conference on Computer Vision*, pages 321–334, London, UK, 1992. Springer-Verlag.
- [FM81] J.T. FEDDEMA et O.R. MITCHELL : Vision-guided servoing with feature-based trajectory generation. *In IEEE Trans. Robotics and Automation*, volume 5(5) : 691–700, 1981.
- [FM89] J.T. FEDDEMA et O.R. MITCHELL : Vision-guided servoing with feature-based trajectory generation. *In IEEE Transactions on Robotics and Automation*, volume 5, n° 5, pages 691–700, October 1989.
- [FT86] O. D. FAUGERAS et G. TOSCANI : The calibration problem for stereo. *In inProc. Conf. Computer Vision and Pattern Recognition*, pages 15–20. Miami,FL, June 1986.
- [Gan99] J. GANGLOFF : *Asservissements visuels rapides d'un robot manipulateur a 6 degrés de liberté*. Thèse de doctorat, Université Louis Pasteur, Janvier 1999.
- [GM78] P.E. GILL et W. MURRAY : Algorithms for the solution of the nonlinear least-squares problem. *SIAM Journal of Numerical Analysis*, 15(5):977–992, 1978.
- [GR70] G.H. GOLUB et C. REINSCH : Singular value decomposition and least squares solution. *In Numer. Math.*, volume 14, pages 403–420, 1970.
- [GV96] G. H. GOLUB et C. F. VAN LOAN : *Matrix Computations*, 3rd ed. Academic Press, Boston, 1996.
- [Has93] K. HASHIMOTO : Visual servoing : Real time control of robot manipulators based on visual sensory feedback. *In World Scientific Series in Robotics and Automated Systems*, volume 7. Singapore, World Scientific Press, 1993.
- [HHC96] S. HUTCHINSON, G. D. HAGER et P.I. CORKE : A tutorial on visual servo control. *In IEEE Transactions on Robotics and Automation*, volume 12, n° 5, pages 651–670, October 1996.
- [Hil62] M. N. HILL : The sea (ideas and observations on progress in the study of the seas). *In In Physical Oceanography*, volume 1. Interscience Publishers, 1962.
- [HRH⁺92] F. HEITGER, L. ROSENTHALER, R. Von Der HEYDT, E. PETERHANS et O. KÜBLER : Simulation of neural contour mechanisms : from simple to endstopped cells. *In Vision Research*, volume 32, n° 5, pages 963–981, 1992.
- [HS88] C. HARRIS et M. STEPHENS : A combined corner and edge detection. *In Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [HT97] G. HAGER et K. TOYAMA : The xvision system : A general-purpose substrate for portable real-time vision applications. *In Computer Vision and Image Understanding*, 1997.
- [HZ83] R.A. HUMMEL et S.W. ZUCKER : On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(3):267–287, May 1983.

- [HZ00] R.I. HARTLEY et A. ZISSERMAN : *Multiple View Geometry in Computer Vision*, page 265. Cambridge University Press, ISBN : 0521623049, 2000.
- [KO94] Takeo KANADE et M. OKUTOMI : A stereo matching algorithm with an adaptive window : theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920 – 932, September 1994.
- [KPG98] R. KOCH, M. POLLEFEYS et L.J. Van GOOL : Multi viewpoint stereo from uncalibrated video sequences. In *ECCV '98 : Proceedings of the 5th European Conference on Computer Vision-Volume I*, pages 55–71, London, UK, 1998. Springer-Verlag.
- [KPG00] Reinhard KOCH, Marc POLLEFEYS et Luc J. Van GOOL : Realistic surface reconstruction of 3d scenes from uncalibrated image sequences. *Journal of Visualization and Computer Animation*, 11(3):115–127, 2000.
- [KR79] L. KITCHEN et A. ROSENFELD : Discrete relaxation for matching relational structures. *IEEE Trans. Systems, Man and Cybernetics*, 9(12):869–874, December 1979.
- [KR82] L. KITCHEN et A. ROSENFELD : Gray level corner detection. *Pattern Recognition Letters*, 1(2):95–102, December 1982.
- [KS00] Kiriakos N. KUTULAKOS et Steven M. SEITZ : A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3):199–218, 2000.
- [KSK⁺98] Reinhard KLETTE, K. SCHLUNS, A. KOSCHAN, Andreas KOSCHAN et Karsten SCHLUNS : *Computer Vision : Three-Dimensional Data from Images*. Springer-Verlag Singapore Pte. Limited, 1998.
- [KZG03] V. KOLMOGOROV, R. ZABIH et S. GORTLER : Generalized multi-camera scene reconstruction using graph cuts. In *In Fourth International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, July 2003.
- [LA04] M.I.A. LOURAKIS et A.A. ARGYROS : The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Rapport technique 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004.
- [Lau95] Aldo LAURENTINI : How far 3d shapes can be understood from 2d silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):188–195, 1995.
- [Lev44] K. LEVENBERG : A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.
- [LH81] H. C. LONGUET-HIGGINS : A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(10):133–135, September 1981.
- [LK81] Bruce D. LUCAS et Takeo KANADE : An iterative image registration technique with an application to stereo vision. pages 674–679, 1981.
- [LL86] P. LIMOZIN-LONG : *Vision stéréoscopique appliquée à la robotique*. Thèse de doctorat, N° 86NICE4051, Université de Nice, France, 1986.

-
- [Low04] D.G. LOWE : Distinctive image features from scale-invariant keypoints. *In International Journal of Computer Vision*, volume 60, 2, pages 91–110, 2004.
- [LT88] R. K. LENZ et R. Y. TSAI : Techniques for calibration of the scale factor and image center for high accuracy 3-d machine vision metrology. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):713–720, 1988.
- [Mal98] E. MALIS : *Contribution à la modélisation et à la commande en asservissement visuel*. Thèse de doctorat, Université de Rennes 1, November 1998.
- [Mal01a] E. MALIS : Vision-based control using different cameras for learning the reference image and for servoing. *In IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 1428–1433. Hawaii, USA, November 2001.
- [Mal01b] E. MALIS : Visual servoing invariant to changes in camera intrinsic parameters. Rapport technique 4309, INRIA, November 2001.
- [Mal02] E. MALIS : A unified approach to model-based and model-free visual servoing. *In European Conference on Computer Vision*, volume 4, pages 433–447. Copenhagen, Denmark, May 2002.
- [Mal04a] E. MALIS : Improving vision-based control using efficient second-order minimization techniques. *In IEEE International Conference on Robotics and Automation*. New Orleans, USA, April 2004.
- [Mal04b] E. MALIS : Visual servoing invariant to changes in camera intrinsic parameters. *In IEEE Transactions on Robotics and Automation*, volume 20(1), pages 72–81, February 2004.
- [Mar63] Donald W. MARQUARDT : An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.
- [Mar04] E. MARCHAND : *Commande d'une caméra réelle ou virtuelle dans des mondes réels ou virtuels*. Habilitation à diriger les recherches, Université de Rennes 1, Mention informatique, November 2004.
- [MB04] E. MALIS et A. BARTOLI : Euclidean reconstruction independent on camera intrinsic parameters. *In IEEE International Conference on Intelligent Robots and Systems*. Sendai, Japan, October 2004.
- [MC00] E. MALIS et F. CHAUMETTE : 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *In International Journal of Computer Vision*, volume 37(1) :79–97, 2000.
- [MCSP02] E. MARCHAND, F. CHAUMETTE, F. SPINDLER et M. PERRIER : Controlling an uninstrumented manipulator by visual servoing. *In The International Journal of Robotics Research*, volume 21(7), pages 635–641, July 2002.
- [MDGD97] P. MARTINET, N. DAUCHER, J. GALLICE et M. DHOME : Robot control using monocular pose estimation. *In Workshop on New Trends In Image-Based Robot Servoing (IROS'97)*, pages 1–12. Grenoble, France, Septembre 1997.

- [MIB04] Iain MATTHEWS, Takahiro ISHIKAWA et Simon BAKER : The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):810 – 815, June 2004.
- [Mor77] H. MORAVEC : Towards automatic visual obstacle avoidance. *In International Joint Conferences on Artificial Intelligence*, page 584, Cambridge, MA, 1977.
- [MS05] Krystian MIKOLAJCZYK et Cordelia SCHMID : A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.
- [OK85] Y. OHTA et T. KANADE : Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:139–154, 1985.
- [Ora01] D. ORAM : Rectification for any epipolar geometry. *In 12th British Machine Vision Conference*, September 2001.
- [PBT07] N. PAPANIKOLOPOULOS, L. BOUDET et O. TOURNAIRE : Automatic man-made object extraction and 3d scene reconstruction from geomatic-images ; is there still a long way to go ? *Urban Remote Sensing Joint Event, 2007*, pages 1–6, 11-13 April 2007.
- [Pes04] N. PESSEL : Camera self-calibration in underwater environment. *In Proceedings of The Fourteenth International Offshore and Polar Engineering Conference*, volume 1, pages 738–745. Toulon, France, May 2004.
- [PG97] M. POLLEFEYS et L. Van GOOL : A stratified approach to metric self-calibration, 1997.
- [PKK93] N. P. PAPANIKOLOPOULOS, P. K. KHOSLA et T. KANADE : Visual tracking of a moving target by a camera mounted on a robot : A combination of control and vision. *In IEEE Transactions on Robotics and Automation*, volume 9(1), Février 1993.
- [PPPD05] L. PÉNARD, N. PAPANIKOLOPOULOS et M. PIERROT-DESEILLIGNY : 3d building facade reconstruction under mesh form from multiple wide angle views. *In Proceedings of the ISPRS Working Group V/4 Workshop “3D-ARCH 2005”*, volume XXXVI. Mestre-Venice, Italy, 2005.
- [RC98] S. ROY et I.J. COX : A maximum-flow formulation of the n-camera stereo correspondence problem. *In Proceedings of 6th International Conference on Computer Vision*, pages 492–499, 1998.
- [RCE89] P. RIVES, F. CHAUMETTE et B. ESPIAU : Visual servoing based on a task function approach. *In First International Symposium on Experimental Robotics*. Montréal, Canada, June 1989.
- [RHZ76] A. ROSENFELD, R. A. HUMMEL et S. ZUCKER : Scene Labeling by Relaxation Operations. *IEEE Trans. Syst., Man and Cybern.*, 6:420–433, 1976.
- [RP05] J. REPKO et M. POLLEFEYS : 3d models from extended uncalibrated video sequences : Addressing key-frame selection and projective drift. *In 3-D Digital Imaging and Modeling*, pages 150–157, 2005.

-
- [S⁺04] P.M. SARRADIN *et al.* : Extreme ecosystem studies in the deep ocean : Technological developments. *In Proceedings of The Fourteenth International Offshore and Polar Engineering Conference*, volume 1, pages 738–745. Toulon, France, May 2004.
- [SB97] Stephen M. SMITH et J. Michael BRADY : Susana new approach to low level image processing. *Int. J. Comput. Vision*, 23(1):45–78, 1997.
- [SBE91] C. SAMSON, M. Le BORGNE et B. ESPIAU : Robot control : The task function approach. *In Oxford Engineering Science Series*, volume 22. Oxford, UK, Clarendon Press, 1991.
- [SBJ90] S. B. SKAAR, W. H. BROCKAN et W. S. JANG : Three-dimensional camera space manipulation. *In Int. Journal of Robotics Research*, volume 9(4) :22–39, Août 1990.
- [SMR96] Cahn Von SEELEN, Ulf M et Bajcsy RUZENA : Adaptive correlation tracking of targets with changing scale. Rapport technique, 1996.
- [SRDT01] Ilya SHLYAKHTER, Max ROZENOER, Julie DORSEY et Seth J. TELLER : Reconstructing 3d tree models from instrumented photographs. *IEEE Computer Graphics and Applications*, 21(3):53–61, 2001.
- [SS00] Heung-Yeung SHUM et Richard SZELISKI : Systems and experiment paper : Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, February 2000.
- [SS02] D. SCHARSTEIN et R.A. SZELISKI : Taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *In International Journal of Computer Vision*, volume 47(1/2/3) :7-42, April–June 2002.
- [SStMcp06] J. SARRAZIN, P.M. SARRADIN et the Momareto cruise PARTICIPANTS : Momareto : a cruise dedicated to the spatio-temporal dynamics and the adaptations of hydrothermal vent fauna on the mid-atlantic ridge. *In InterRidge News*, volume 15, pages 24–33, 2006.
- [SSZ01] D. SCHARSTEIN, R. SZELISKI et R. ZABIH : A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, 2001.
- [ST94] Jianbo SHI et Carlo TOMASI : Good features to track. *In Int. Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
- [STA98] G. SUBSOL, J.-Ph. THIRION et N. AYACHE : A scheme for automatically building 3D morphometric anatomical atlases : application to a skull atlas. *Medical Image Analysis*, 2(1):37–60, 1998.
- [Stu97a] P. STURM : Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. *In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1100–1105, 1997.
- [Stu97b] Peter STURM : *Vision 3D non calibrée : contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage*. Thèse de doctorat, Institut National Polytechnique de Grenoble, December 1997.

- [SW80] A.C. SANDERSON et L.E. WEISS : Image based visual servo control using relational graph error signal. *In In Proc. Int. Conf. Cybern. and Soc.*, pages 1074–1077. Cambridge, MA, October 1980.
- [TB97] Lloyd N. TREFETHEN et David BAU III : *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
- [TFZ98] P. TORR, A. W. FITZGIBBON et A. ZISSERMAN : Maintaining multiple motion model hypotheses over many views to recover matching and structure. *In IEEE International Conference on Computer Vision*, pages 485–491. Bombay, India, January 1998.
- [Thi96] Jean-Philippe THIRION : New feature points based on geometric invariants for 3d image registration. *Int. J. Comput. Vision*, 18(2):121–137, 1996.
- [TK91] Carlo TOMASI et Takeo KANADE : Detection and tracking of point features. Rapport technique CMU-CS-91-132, Carnegie Mellon University, April 1991.
- [TK92] C. TOMASI et T. KANADE : Shape and motion from image streams under orthography : a factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.
- [TMHF00] B. TRIGGS, P. MCLAUCHLAN, R. HARTLEY et A. FITZGIBBON : Bundle adjustment – a modern synthesis. *In B. TRIGGS, A. ZISSERMAN et R. SZELISKI, éditeurs : Vision Algorithms : Theory and Practice*, volume 1883 de *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.
- [Tri96] B. TRIGGS : Factorization methods for projective structure and motion, 1996.
- [TS04] P. TISSAINAYAGAM et D. SUTER : Assessing the performance of corner detectors for point feature tracking applications. *Image and Vision Computing*, 22(8):663–679, 2004.
- [TSSN97] M. TONKO, J. SHURMANN, K. SCHAFER et H.-H. NAGEL : Visually servoed gripping of a used car battery. *In IEEE Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 49–54. Grenoble, France, Septembre 1997.
- [WCG96] Wilson W.J., Williams Hulls C.C. et Bell G.S. : Relative end-effector control using cartesian position based visual servoing. *In IEEE Transactions on Robotics and Automation*, volume 12(5), pages 684–696, 1996.
- [Wei84] L. WEISS : *Dynamic visual servo control of robots : an adaptative image-based approach*. Thèse de doctorat, Carnegie-Mellon University, 1984.
- [WF97] P. WHAITE et F. P. FERRIE : Autonomous exploration : Driven by uncertainty. *In IEEE Transactions Pattern Analysis and Machine Intelligence*, volume 19(3), pages 193–205, 1997.
- [WSN87] L.E. WEIS, C. SANDERSON et C.P. NEURAN : Dynamic sensor-based trajectory generation. *In IEEE Journal of Robotics and Automation*, volume RA-3, n° 5, pages 404–417, October 1987.
- [ZCS02] Li ZHANG, Brian CURLESS et Steven M. SEITZ : Rapid shape acquisition using color structured light and multi-pass dynamic programming. *In The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, June 2002.

-
- [Zha00] Zhengyou ZHANG : A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.

Résumé

L'objectif de cette thèse est de proposer une méthodologie complète de reconstruction 3D d'objets sous-marins naturels, améliorée par une nouvelle méthode d'acquisition afin de permettre des mesures quantitatives. Il a d'abord fallu prendre en compte les différents problèmes liés au milieu sous-marin profond ; la contrainte principale est que le système utilisé pour faire l'acquisition des images doit être contrôlé à des profondeurs très importantes, jusqu'à 6000 mètres, à l'aide d'un véhicule positionné sur le fond. Ainsi, une méthode permettant l'acquisition automatique d'images a été développée, adaptée à tout type d'objet sous-marin de faible échelle (environ $1m^3$). L'acquisition d'image est réalisée avec un système de stéréovision contrôlé par un bras manipulateur. La méthode que nous proposons permet de connaître les paramètres extrinsèques des caméras du système de vision, par le suivi d'une trajectoire définie par la géométrie de la tête stéréo. Ainsi, la trajectoire est générée par le déplacement d'une caméra sur la position de l'autre caméra par asservissement visuel. Avec cette méthode, nous pouvons enregistrer des images à intervalles réguliers directement liés à la géométrie de la tête stéréo. Ensuite, le modèle 3D de l'objet sous-marin est calculé à partir des images collectées et des paramètres des caméras. Le résultat final est une reconstruction 3D dense avec un plaquage de texture, qui permet de faire des mesures métriques.

Mots-clés: métrologie 3D, vision par ordinateur, stéréovision, asservissement visuel, trajectoire d'acquisition, reconstruction 3D.

Abstract

The aim of this study is to propose a complete 3-dimension reconstruction method of natural submarine objects improved by a new acquisition method for quantitative measures, which can be used in operational conditions. First, it was necessary to take into account the various problems connected with the deep sea environment ; the main constraint is that the system used to collect images must be manipulated at very important depths, up to 6000 meters by an underwater vehicle positioned on the sea floor. Thus, a method allowing the automatic acquisition of images was developed, adapted to any type of small-scale submarine object (approximately $1m^3$). The image acquisition is performed with a stereovision system operated by a manipulator arm. The method that we propose enables us to know extrinsic camera parameters by following a specific trajectory defined by the geometry of a stereo rig. Indeed, the trajectory is generated by the displacement of one camera onto the position of the other one by visual servoing. With this method, we can register images at regular intervals directly linked to the geometry of the stereo rig. Then, the 3D model of the underwater object is calculated from the collected images and camera parameters. The final result is a dense 3D reconstruction with texture mapping that enables metric measures.

Keywords: 3D metrology, computer vision, stereovision system, visual servoing, camera trajectory, 3D reconstruction.