

N° d'ordre : 3837

THÈSE

présentée

devant l'Université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention INFORMATIQUE

par

Claire DUNE

Équipe d'accueil : IRISA - Équipe projet LAGADIC
École Doctorale : Matisse
Composante universitaire : IFSIC

Titre de la thèse :

**Localisation et caractérisation d'objets inconnus
à partir d'informations visuelles :**
vers une saisie intuitive
pour les personnes en situation de handicap

Soutenue le 30 avril 2009 devant la commission d'examen

Luce	MORIN	Présidente du Jury
Étienne	COLLE	Rapporteurs
Michel	DEVY	
Frank	FERRIE	Examineurs
Sylvie	LAMY	
Éric	MARCHAND	Directeur de Thèse

à Petit Picasso, Mémé Doudou et aux ingénieurs yeuteurs

Prenez un brin de Lagadic, trois pincées de sel de la Region Bretagne, quelques atomes du CEA LIST, un clic de souris, un crabe japonais, une cloche, une bouffée d'air frais, capturé en altitude, là haut dans les Alpes, une bonne dose de café, beaucoup d'amitiés, un vieux cuir Mikasa et mélangez bien durant un peu plus de trois années, de préférence à l'aide d'un bras Manus. Laissez reposer le mémoire obtenu un mois ou deux puis le servir chaud accompagné de quelques crêpes.

Mes remerciements vont d'abord aux membres du jury. Je remercie Luce Morin, Professeur à l'INSA de Rennes de m'avoir fait l'honneur de présider le Jury. Je remercie Etienne Colle, Professeur à l'Université d'Evry et Michel Devy, Directeur de recherche au LAAS d'avoir accepté la charge de rapporteurs en goûtant les deux premières parts. Merci à Frank Ferrie, Professor à Mc Gill d'avoir traversé l'atlantique le temps d'un long week end pour prendre part au Jury.

Je remercie la Bourse Bretagne et le CEA pour leur soutien financier.

Merci à Francois et Yvan pour m'avoir accueillie au sein de leurs équipes à Rennes et à Fontenay aux Roses et pour m'avoir donné le support matériel et l'encadrement nécessaire pour mener cette thèse à son terme. Merci pour votre attention, votre écoute et la confiance que vous m'avez témoignée. Merci Francois pour les pas de danses à Domagné et merci Yvan pour les plateaux et les smashé à Fontenay.

Un Merci tout particulier au (presque) Professeur Eric Marchand qui a su diriger ces travaux de Rennes à Paris, et dispenser de précieux conseils tout en me donnant l'autonomie suffisante pour que je puisse m'approprier ce sujet de thèse. Merci aussi pour ta disponibilité, surtout à la veille de certaines dates limites.

Merci Sylvie, pour avoir repris l'encadrement de cette thèse, côté CEA, à quelques mois de son terme, aux premières lignes de ce manuscrit. Merci pour ta bonne humeur, ton optimisme et le coup de pouce dans la gestion du temps de ces derniers mois à Paris qui m'a permis de ne pas perdre pieds.

Il ne seront jamais cités officiellement au nombre des encadrants de cette thèse, pourtant ils y ont grandement contribué.

Merci Nicolas et Anthony pour avoir accompagné mes premiers pas à Rennes puis à Paris. Merci pour les heures tardives passées à lire, corriger et relire mes articles. Merci pour vos conseils sur l'après thèse. Merci pour tout le temps que vous m'avez accordée pour me donner conseils scientifiques et techniques ou encore sur l'après thèse. Merci Anthony pour le travail de préparation que tu as fait sur les codes du projet Aviso à la suite duquel cette thèse s'est inscrite. Merci Nicolas pour m'avoir permis de visiter l'AIST ou j'aurai, un peu grâce à toi, l'opportunité de faire un post-doctorat. Merci aussi Anthony, pour les longues discussions le matin autour du café qui réveille et pour les tartares chez Tonton.

Merci Christophe Collewet pour ton travail sur les contours actifs et pour ton aide sur les parties d'optimisation numérique. Merci pour tes relectures et tes encouragements. Bonne recherche et bon vent dans l'équipe Fluminance!

Merci ViSP. Merci Fabien, Francois, Eric, Anthony, Nicolas pour votre travail sur la librairie ViSP qui nous permet de gagner beaucoup de temps lorsque l'on souhaite mettre en place des essais. Merci Fabien pour le maintien des robots de Lagadic. Ton travail est indispensable aux doctorants et nous permet de nous consacrer aux problématiques centrales de nos thèses. Merci pour ta gentillesse, ta disponibilité et les heures passées à débayer scripts et morceaux de codes.

Merci Celine, merci Anne pour nous rendre la vie facile et vous occuper de tous les à-côtés de la thèse. Grâce à vous, l'administratif devient un jeu d'enfant!

Merci Geneviève pour tes conseils précieux en anglais scientifique.

Je remercie vivement les très nombreux relecteurs de ce manuscrit qui ont tenté à maintes reprises de bouter les fautes de français hors de ce mémoire. J'ai perdu le compte à 25 personnes... Collègues roboticiens, famille, coéquipières volleyeuses, je vous ai mis à rude contribution. Merci en particulier

à Caroline, que j'ai eu la chance d'encadrer durant son stage de Master et qui a sans doute relu à elle seule plus que tous les autres réunis les pages de ce manuscrit. Merci à Ana, Céline et Anthony pour votre résistance face aux relectures multiples des moutures successives des mêmes chapitres. Merci à Eric et Sylvie qui ont contribué au delà de leurs prérogatives d'encadrants à la correction de ce mémoire. Merci à mes parents et beaux parents qui ont relu ce texte crypté d'un bout à l'autre. Merci à Didier qui a vérifié toutes les lignes mathématiques de cette thèse. Merci à Nicolas Biard pour sa relecture de la préface.

Merci à tous ceux qui m'ont aidé à préparer la soutenance en assistant assiduellement aux répétitions : merci Andrea, Romeo, Jean, Caroline, Olivier, Fabien, Amaury, Sylvie, Eric, François, Christophe, et un merci tout particulier à Ana Charpentier, Nicolas Wiest Daesle et Ivan Laptev qui ont éclairé d'un regard extérieur ces présoutenances.

Sans eux, la sauce ne prend pas, je voudrais remercier mes collègues de l'équipe Lagadic et du Laboratoire de Téléopération et Cobotique. Merci Andrea pour avoir résisté au souffle des dernières heures de la thèse. Merci Odile pour les tartes au citron. Merci Muriel pour tes conseils et la sauce à la prune. Merci Romeo pour ton sourire, les midis sportifs, le coupé-décalé, les happy hours. Merci Alexandre pour l'histoire des chaussures et du 'bloody steak'. Merci Amaury, (aka Obélix) pour avoir filmé La séquence, tes plantes vertes et pour l'initiation à l'ultimate fighting. Merci Céline (aka Falbala) pour le pommier d'amour et les nuits niçoises. Merci Mohammed pour les épices ramenées de loin, tes cafés égyptiens, tes photos du 6 juin. Merci Olivier pour les discussions d'autom et pour m'avoir appris le nom qu'à la petite fenêtre triangulaire à l'avant d'une voiture. Merci Sinisa pour les sifts et les sorties à Dol et Dinard. Merci Albert pour le nouveau vocabulaire et le saut de l'ange. Merci Gérard pour les cookies et les fonds d'écrans. Merci Franck de ta gentillesse et pour l'accroche sous marine. Merci FX pour tes conseils sur tao et la traduction de ses warnings. Merci Aurélien pour la souris magique. Merci Fred et Laurent pour vos conseils. Merci Don Camillo pour les crabes. Merci Julie, Caroline, Joseph et Alexis pour Don Camillo. Merci Greg pour la lecture quotidienne et les astuces du CEA. Merci Bruno, Xavier, Jean-Baptiste et Matthieu pour l'ambiance du bureau 17, la petite balle, les gummies et le hockey unique là, juste devant l'accueil du CEA. Merci Max pour le casque anti bruit. Merci Nicolas pour les drones aux dessus de Fontenay. Merci Ryuta pour les premiers mots en japonais et Xiang pour leurs traduction chinoise. Merci Tao, Ravo, Emilio.

Hors les murs du labo, pour l'autre vie et toutes les petites choses qui la composent, je voudrais remercier mes amis et ma famille. Merci pour toutes les parenthèses que vous m'avez offertes, les soirées sushis, la fabrication de yourte, les balades à cheval, les sorties escalades, la chasse à l'araignée, les retours à la frontal, la construction d'igloo, les greg girls en pré-nationnale. Merci de m'avoir emmenée avec vous et d'avoir fait voyager cette thèse aux quatre coins de la France et à l'autre bout du monde.

Et pour tout le reste, merci François. Cette thèse est aussi la tienne.

Table des matières

Table des matières	1
1 L'assistance robotisée aux personnes en situation de handicap	9
1.1 L'assistance robotisée à la saisie	9
1.1.1 Une discipline de la robotique médicale	10
1.1.2 Les besoins des personnes en perte d'autonomie en termes d'assistance robotisée	13
1.1.3 Quels critères doit satisfaire une aide robotisée destinée aux personnes en perte d'autonomie?	16
1.2 Les systèmes d'aide à la saisie	19
1.2.1 La saisie d'objet sans perception de l'environnement.	19
1.2.2 La saisie d'objets en environnement limité	24
1.2.3 La saisie d'objets en environnement humain	28
1.3 Proposition pour un outil de saisie contrôlé par la vision	32
1.3.1 Présentation générale	32
1.3.2 La perception de l'environnement	35
1.3.3 La désignation de l'objet	38
1.3.4 Le scénario de saisie	39
1.3.5 Les contributions de cette thèse	41
I <i>One Click Focus</i> : estimation de la position d'un objet inconnu à partir d'un clic	43
2 Représentation géométrique	47
2.1 Rappels mathématiques et géométriques	47
2.1.1 De l'espace euclidien à l'espace projectif.	47
2.1.2 Le modèle sténopé	49
2.1.3 Système à deux caméras	55
2.2 Un outil de saisie guidé par la vision	57
2.2.1 Repères et Notations	57
2.2.2 Stratégie de localisation.	59
3 Une commande visuelle basée sur la géométrie épipolaire	61
3.1 Loi de commande par asservissement visuel	63
3.1.1 Le formalisme de l'asservissement visuel	63
3.1.2 De la redondance en asservissement visuel	65
3.2 Loi de commande appliquée à la caméra embarquée pour localiser l'objet	67
3.2.1 Tâche principale : centrage de la ligne	68

3.2.2	Tâche secondaire : parcours de la ligne	69
3.2.3	Tâche à dimension variable : évitement des butées	70
3.3	Résultats expérimentaux	72
3.3.1	Parcours de la ligne de vue	73
3.3.2	Parcours de la ligne de vue en évitant les butées	77
3.3.3	Suivi d'un objet en mouvement	77
4	Localisation de l'objet sur une ligne de vue	85
4.1	Un problème de reconnaissance d'apparence	85
4.2	Méthode de reconnaissance robuste	87
4.2.1	Invariants robustes aux changements d'échelle et d'orientation	87
4.2.2	Mise en correspondance des descripteurs SIFT	87
4.2.3	Élimination des faux positifs.	90
4.3	Localisation d'un objet inconnu	90
4.3.1	Estimation de la vraisemblance associée à une vue	91
4.3.2	Fusion des données sur l'ensemble des vues.	101
4.4	Résultats expérimentaux	103
II Caractérisation de la forme d'un objet inconnu et exploration par vision active		111
5	Types d'objets et stratégie de saisie	115
5.1	Hypothèses générales	115
5.2	La stratégie de saisie	116
5.3	Modélisation retenue	118
5.4	Acquisition des données	118
6	Modélisation d'un objet par une quadrique	121
6.1	Définitions	121
6.2	Calcul des informations nécessaires à la saisie	123
6.2.1	Centre d'une quadrique	123
6.2.2	Axes principaux et dimensions	124
6.3	Propriétés projectives des quadriques	125
6.3.1	Dans l'espace projectif.	126
6.3.2	Dans l'espace dual.	127
7	Estimation des paramètres d'une quadrique	129
7.1	Estimation des paramètres d'une conique.	130
7.2	Une reconstruction qui exploite les propriétés de l'espace dual	132
7.3	Une reconstruction basée sur la relation directe entre une quadrique et ses projections	133
7.4	Une reconstruction robuste	138
7.5	Résultats de simulation	139
7.5.1	Reconstruction non bruitée d'une quadrique	140
7.5.2	Reconstruction bruitée d'une quadrique	141
7.5.3	Reconstruction d'une quadrique à partir de vues proches	146
7.5.4	Reconstruction d'un objet de forme quelconque	146
7.6	Conclusion	151

8	Détection de l'objet par une méthode de contours actifs	155
8.1	Méthodes existantes	155
8.2	Détection et suivi de contours par contours actifs	158
8.2.1	Principe des contours actifs	158
8.2.2	Méthode développée	159
8.2.3	Résultats expérimentaux	161
8.3	Initialisation des contours actifs en un <i>clic</i>	161
8.3.1	Initialisation à partir d'un point pour la détection d'un objet uniforme	164
8.3.2	Initialisation et suivi en utilisant une boite englobante	164
8.3.3	Évaluation de l'échelle caractéristique de l'objet	164
8.3.4	Orientation de l'objet	169
8.4	Conclusion	172
9	Sélectionner les meilleures vues : une méthode de vision active	175
9.1	Information contenue dans une vue	176
9.1.1	Incertitude sur les paramètres	176
9.1.2	Critère de sélection de la vue suivante	177
9.2	Calcul de la position suivante	178
9.3	Résultats Expérimentaux	179
9.4	Conclusion	183
A	Rappel des notations	187
B	Le bras MANUS	189
C	Hypothèses	193
D	Démonstrations	195
	Table des figures	207

Problématique

S'il semble prématuré de parler d'intelligence robotique, les avancées technologiques de ces dernières années permettent déjà à de petits systèmes robotiques de trouver leur place chez les particuliers. Parmi eux, on peut citer les robots jouets, les systèmes de surveillance, les aspirateurs, les robots nettoyeurs de piscine ou encore les tondeuses autonomes. Selon une étude de la commission économique de l'ONU, il y avait en 2007 plus de 8 millions de robots domestiques dans le monde. Début 2007, Bill Gates, se joint à l'élan asiatique et lance sa campagne "*Un robot par foyer*" (*A Robot in Every Home*) [Gates 07] en comparant l'évolution actuelle de la robotique à celle des ordinateurs dans les années 70. Il y expose sa vision de la robotique personnelle : plutôt que de concevoir un robot pouvant réaliser une grande variété de tâches, il imagine un habitat domotisé permettant de gérer un ensemble de robots spécialisés (voir la figure 1).

Aujourd'hui, 10% de la population mondiale a plus de 60 ans et cette proportion devrait doubler d'ici à 2050, se concentrant massivement dans les pays industrialisés [Mann 05]. Cette population en perte d'autonomie s'ajoutera à la population de personnes handicapées nécessitant une assistance quotidienne et surpassera largement le potentiel d'accueil des centres spécialisés d'ici quelques années. Pour palier à ce problème d'accueil, une solution envisagée consiste à apporter une aide adaptée qui permettrait aux personnes âgées ou handicapées de rester chez elles le plus longtemps possible. La perte d'autonomie qui les conduit dans les instituts ou les maisons de retraite serait alors repoussée. Cette aide peut être apportée par des aides soignants et par une modification de l'infrastructure (domicile, centre spécialisé). Cependant, le nombre d'aides soignants a tendance à se stabiliser et sera rapidement insuffisant. Une aide robotisée complémentaire à cette aide humaine permettrait de diminuer le temps de présence de la tierce personne. Au Japon, où 19% de personnes ont aujourd'hui plus de 65 ans, les assistants robotiques sont déjà une réalité. En 2015 il est prévu qu'un quart des foyers nippons sera équipé d'assistants robotiques domestiques, objets communicants mobiles allant de l'accompagnement psychologique à la télésurveillance médicale.

Parmi les fonctionnalités que devrait compter l'assistant robotique, certaines demandes sont partagées par les personnes âgées et les personnes handicapées, comme la possibilité de saisir et de manipuler des objets du quotidien afin de pouvoir réaliser des tâches simples en l'absence d'aides soignants. Pour répondre à ce besoin plusieurs solutions ont été proposées depuis le début des années 70. Les premières tentatives utilisaient des bras industriels et permettaient de saisir des objets préalablement positionnés par une personne valide à des emplacements dédiés et connus du système. L'environnement était alors structuré et contraint. Par la suite, des systèmes portables et légers, pouvant être fixés sur un fauteuil roulant ou sur des plateformes mobiles ont été développés, ouvrant sur de nouveaux challenges, tels que la saisie d'objets dont l'emplacement n'est pas connu et dans un environnement qui n'est pas contraint.

Pour répondre à ces nouveaux besoins, trois modes de commande du système ont été envisagés. Une première solution est de n'attribuer aucune autonomie au robot et d'utiliser un mode de commande manuel ou téléopéré. Le système n'a aucune perception de l'environnement. L'utilisateur perçoit, interprète, décide et le robot "agit". Ses mouvements sont commandés via un périphérique



FIG. 1 – Une vision de la robotique personnelle [Gates 07] : plusieurs robots spécialisés sont affectés à des tâches spécifiques dans une maison domotisée qui coordonne l'ensemble.

adapté au handicap de l'utilisateur. Ce dernier est alors sollicité à chaque étape de la saisie ; ce qui lui demande un certain entraînement, de la dextérité et de l'attention. Selon le degré du handicap, une telle implication n'est pas toujours souhaitable ou possible.

À l'inverse, une seconde voie d'étude tend à donner toute autonomie au système. L'utilisateur est exclu de la boucle de commande, son action se limitant à donner un ordre de haut niveau pour déclencher l'action. À partir de connaissances préalablement acquises et de sa perception de l'environnement, le robot "interprète", "décide" et "agit". Le système est complexe et peut demander des temps de calculs importants pour parvenir à une prise de décision. D'autre part il est difficile de garantir sa robustesse. En effet, alors que l'être humain peut s'adapter à des situations nouvelles, les systèmes actuels sont encore loin d'égaliser ses facultés d'abstraction et peuvent échouer s'ils sont confrontés à des situations qui ne sont pas répertoriées dans leur base de connaissances.

La solution se trouve peut être dans un partage judicieux de l'autonomie, en développant un mode de commande dit "semi automatique" ou "partagé". Dans ce cas, le robot dispose d'une part d'autonomie qui soulage l'utilisateur d'une partie de la commande. L'objectif est de minimiser la fatigue engendrée par l'utilisation du système tout en limitant la complexité du robot. L'utilisateur et le robot perçoivent l'environnement et l'interprètent. Cependant les décisions de haut niveau sont laissées à l'humain et l'action au système. En laissant en permanence à l'utilisateur la main sur le système, il est plus facile de garantir une certaine robustesse et sa sécurité. L'utilisateur peut interrompre à tout moment le déplacement du robot ou le modifier s'il le juge inapproprié. D'autre part, certaines tâches, comme par exemple l'identification d'objets ou encore leur utilisation, est immédiate pour l'utilisateur alors qu'elle est très complexe pour le système. La difficulté est alors de trouver un compromis entre sollicitation de l'utilisateur et complexité du système.

Le travail présenté dans cette thèse s'inscrit dans le cadre d'un projet pour l'assistance aux personnes handicapées mené conjointement par le CEA-LIST de Fontenay-aux-Roses (Laboratoire de Téléoopération et de Cobotique du Commissariat à l'Énergie Atomique), l'association APPROCHE (Association pour la promotion des nouvelles technologies en faveur des personnes en perte d'autonomie) et l'AFM (Association Française contre les Myopathies). Ce projet tente d'étendre l'utilisation de robots assistants à l'ensemble des personnes en situation de handicap et s'attache plus particulièrement aux tâches de saisie et de manipulation d'objets pour redonner à ces personnes une certaine autonomie dans les activités de la vie quotidienne : saisir un objet sur un meuble, ramasser un objet tombé au sol, se nourrir, etc. Le projet a pour but de simplifier l'interface de commande et permettre au système de saisir n'importe quel objet rencontré par une personne handicapée, dans son logement aussi bien que dans les commerces ou les espaces publics.

Dans ce contexte, l'objectif est de proposer des outils pour saisir des objets génériques, tirés du quotidien, en utilisant uniquement des informations fournies en ligne par l'utilisateur et des caméras montées sur le système robotique. L'environnement est ouvert, c'est-à-dire qu'il n'est pas limité à un nombre d'objets prédéfinis dont les caractéristiques seraient stockées dans une base de données. D'autre part, nous nous sommes attachés à réduire l'information donnée par l'utilisateur à une unique désignation de l'objet sur un écran affichant une vue de la scène. La commande du bras consiste à saisir l'objet désigné et à le ramener à proximité de la personne.

La tâche de saisie peut être décomposée en quatre grandes étapes qui structurent ce manuscrit : la détection de l'objet, sa localisation, l'approche grossière et enfin la saisie. La détection de l'objet est en fait laissée à l'utilisateur qui transmet l'information au système par un *clic* sur l'objet à saisir. La localisation de l'objet à partir du *clic* et la représentation de l'objet permettant de positionner les mors de la pince pour saisir correctement l'objet forment le cœur de ce manuscrit. Enfin, la fermeture de la pince est déclenchée lorsque l'objet coupe une barrière optique montée sur la pince.

L'assistance robotisée aux personnes en situation de handicap

Cette thèse s'inscrit dans le prolongement d'un projet de robotique d'assistance à la saisie d'objets pour les personnes handicapées, mené au CEA-LIST en partenariat avec les associations APPROCHE et AFM. Ce premier chapitre présente le contexte applicatif et les motivations de cette thèse.

Dans un premier temps, les besoins des personnes handicapées en terme d'assistance seront présentés et en particulier il sera montré que la saisie et la manipulation d'objets font partie des tâches prioritaires qu'elles souhaiteraient pouvoir réaliser pour recouvrer une certaine autonomie au quotidien. Une aide robotisée peut répondre à ces besoins à la condition qu'elle respecte certains critères qui ont été établis par les équipes cliniques et les patients au cours de campagnes de tests sur des systèmes réels. Ensuite, une section sera consacrée aux systèmes d'assistance à la saisie existants. Ces systèmes seront classifiés en tenant compte des hypothèses de travail qu'ils utilisent et des stratégies adoptées pour saisir des objets. Puis nous présenterons les choix qui ont conduit au système sur lequel nous travaillons : un bras déjà commercialisé qui peut être monté sur un fauteuil ou sur une plateforme mobile et équipé d'un système de vision.

1.1 L'assistance robotisée à la saisie

"Un senior pourra demander au robot qui comprendra le langage humain, de commander de la nourriture au restaurant d'en face par exemple. Le robot ira chercher la commande et servira le dîner. Il débarrassera et fera la vaisselle. Il s'occupera même de la santé de son maître car il prendra ses pulsations cardiaques et sa tension et informera son médecin. Il pourra aussi donner l'alerte si son propriétaire tombe et ne peut se relever. Par ailleurs, il fera aussi office de secrétaire et d'ami proche. Ainsi, ces robots serviront aussi à lutter contre le sentiment de solitude souvent ressenti par les personnes âgées. On voit donc que les robots d'assistance et de compagnie pour les personnes âgées ne relèvent plus de la science-fiction et commenceront, d'ici une dizaine d'années, à devenir d'indispensables et précieux auxiliaires de vie auprès de nos aînés, tant à leur domicile qu'en institution. Reste cependant une grande inconnue : le degré d'acceptabilité dont feront preuve les personnes âgées à l'égard de ces robots qui les accompagneront dans leur vie quotidienne.", écrit le sénateur du Rhône

René Tréguet.¹

Un robot pour assister les personnes âgées², tel est le rêve du sénateur René Tréguet pour palier au déficit d'accueil dans les centres spécialisés. Si les robots compagnons qu'il décrit peuvent rendre une certaine autonomie à des personnes âgées ou en situation de handicap, il ne sont pas encore au seuil de nos maisons.

Dans un premier temps, cette partie situe la robotique d'assistance au sein de la robotique médicale. Nous montrerons ensuite quels sont les besoins exprimés par les personnes en situation de handicap et les équipes médicales. Finalement, nous dresserons la liste des critères qui permettent d'évaluer un bon système d'assistance.

1.1.1 Une discipline de la robotique médicale

Depuis les années 60, la robotique a su trouver sa place dans le domaine de l'assistance à la personne et plus largement dans le milieu médical, donnant naissance à de nouveaux domaines d'application (voir figure 1.1).

Le premier champ applicatif est la robotique chirurgicale qui améliore considérablement le confort du patient et la précision du médecin. Depuis la fin des années 90, des systèmes tels que les robots Zeus et Da Vinci assistent le geste médical dans la chirurgie mini-invasive. Sur certaines opérations intra-abdominales, les risques de complications qui étaient de 19% en chirurgie traditionnelle tombent à 6%. Les patients reviennent à une activité normale au bout de 27 jours au lieu de 52 auparavant. Le seul frein à l'utilisation de ces systèmes est l'augmentation du coût des opérations qu'ils engendrent.

Parallèlement à la robotique chirurgicale, des simulateurs ont été développés pour la formation à l'utilisation de ces nouveaux outils mais aussi pour simuler le comportement des patients. Des centres de formation du type de celui de l'IRCAD à Strasbourg, permettent aux chirurgiens de se former en chirurgie mini-invasive en utilisant des bras à retour d'effort. Au Japon, des simulateurs de patients ont été conçus pour l'entraînement aux soins dentaires. Ces équipements permettent aux praticiens de réaliser les soins dans des conditions proches des conditions réelles tout en ayant un retour quantitatif sur les soins administrés, ce qui leur permet éventuellement de corriger leurs gestes.

La robotique d'assistance fait partie de la robotique de réhabilitation qui désigne tous les systèmes qui permettent de compenser ou d'améliorer une fonctionnalité manquante ou déficiente du corps humain. En plus des systèmes d'assistance, elle englobe également les robots orthèses (aussi appelés exosquelettes), les prothèses robotisées et les systèmes de rééducation (voir figure 1.1).

Une prothèse robotique est un ensemble mécanique actionné qui se substitue à une partie manquante du corps humain. Ces systèmes sont utilisés le plus souvent pour redonner des capacités de mobilité ou de manipulation lorsqu'un membre est manquant. La figure 1.2 présente deux prothèses récemment développées. Actuellement, la recherche tente de développer des interfaces qui permettraient une communication directe entre les actionneurs et les capteurs du robot, et le système nerveux du corps humain.

Les orthèses sont des squelettes externes qui renforcent une articulation ou un muscle inefficace. Elles sont utilisées pour palier à une déficience permanente ou pour servir à amplifier la force du corps humain pour porter des charges lourdes. Elles sont également utilisées en rééducation, comme les orthèses Able du CEA et ARmin de l'ETH Zurich (voir figure 1.3). La recherche s'intéresse principalement à l'amélioration de la morphologie du robot qui est intimement lié au corps humain. La conception d'orthèses et de prothèses demande d'accorder un soin particulier au *design* du système

¹Extrait du site de la société Altavis, société de conseil auprès des collectivités territoriales et des entreprises de haute technologie dans les domaines de la prospective, de la stratégie, de l'innovation, des nouvelles technologies de l'information et de l'aménagement du territoire (<http://www.altavis.fr>).

²Les personnes âgées présentent une perte d'autonomie qui peut parfois demander les mêmes outils que des personnes en situation de handicap. Dans ce chapitre, nous parlerons donc à la fois des populations vieillissantes et des personnes en situation de handicap.

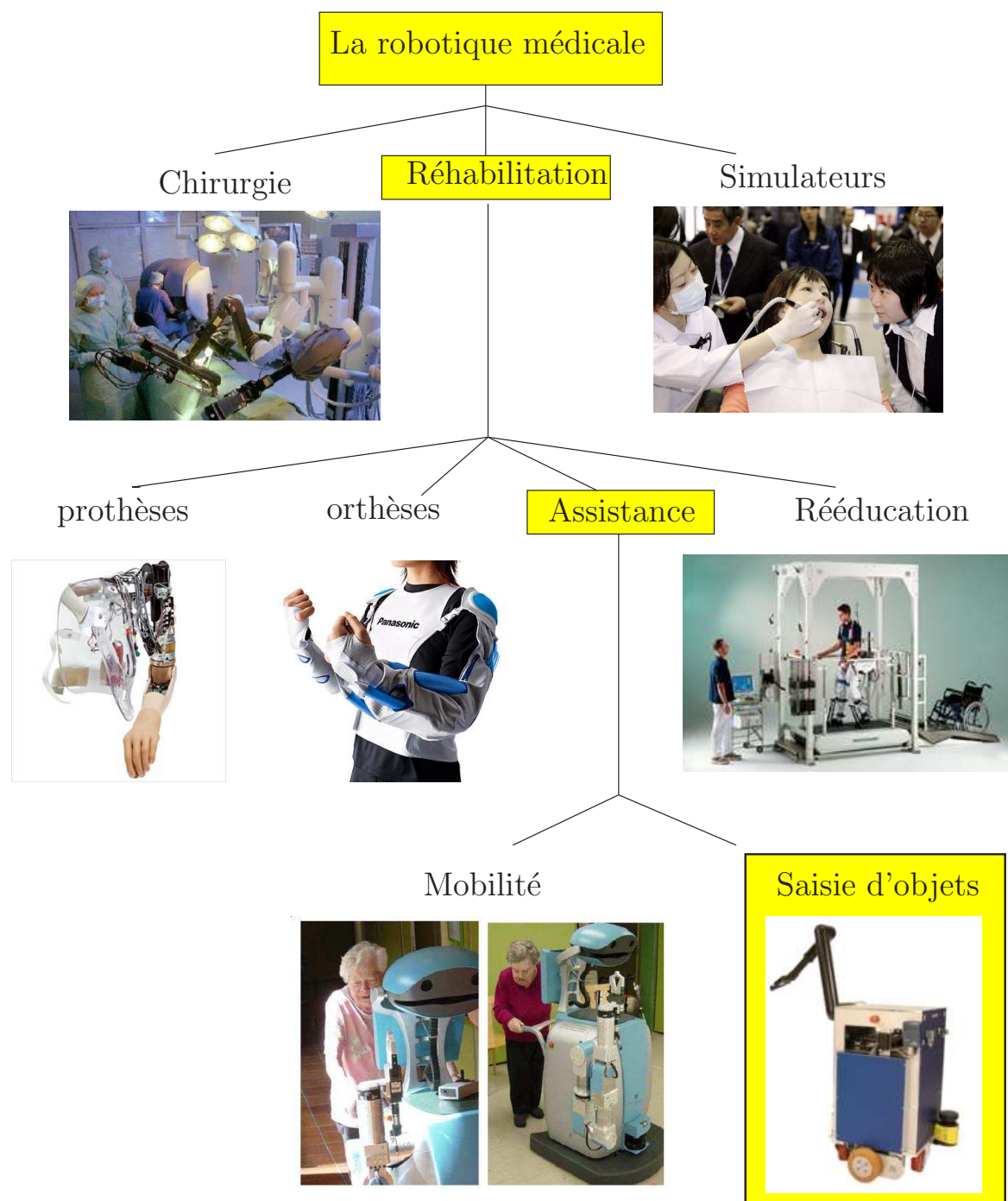


FIG. 1.1 – La robotique médicale. Trois familles de systèmes robotiques peuvent être distinguées, dans le domaine de la chirurgie, de la réhabilitation et de la simulation d'acte médicaux. Parmi les robots de réhabilitation, on trouve les prothèses, les orthèses (ou exosquelettes), les robots d'assistance à la saisie ou à la marche et les robots de rééducation. La limite entre ces catégories est floue et il est courant de considérer que les orthèses et les prothèses sont des robots d'assistance.



FIG. 1.2 – À gauche, une prothèse de bras, à droite, une prothèse de cheville et de pied.

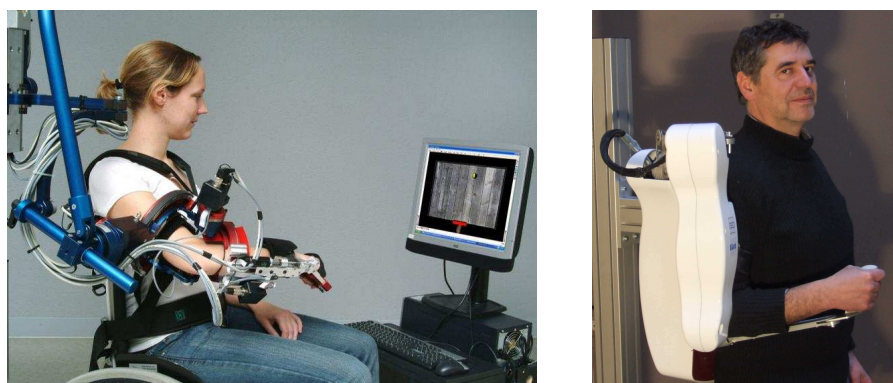


FIG. 1.3 – À gauche, l'orthèse ARmin de l'ETHZ à droite l'orthèse Able du CEA-LIST. Ces orthèses sont utilisées pour la rééducation du membre supérieur.

pour qu'il soit léger et que l'encombrement soit minimal. Ces systèmes sont portés par la personne et sont téléopérés.

Dans le domaine de la rééducation, l'objectif est de produire des systèmes permettant de faire travailler des segments du corps humain indépendamment les uns des autres et de pouvoir évaluer quantitativement l'évolution de l'état de santé du patient. Le Lokomat est présenté sur la figure 1.4 et illustre la branche rééducation de la figure 1.1. C'est un tapis roulant équipé d'une potence qui permet de soutenir le patient. Il permet de rééduquer la marche tout en permettant de limiter les complications liées à une position assise prolongée.

Enfin, le dernier domaine de la robotique de réhabilitation est la robotique d'assistance. Ce champ applicatif peut être divisé en deux sous catégories : les aides à la mobilité et à la saisie. Malgré la richesse des développements de ces dernières années, les solutions proposées ne parviennent pas à être diffusées auprès des personnes en situation de Handicap. Actuellement, les aides techniques à la saisie les plus utilisées sont des prothèses ou des orthèses qui sont entièrement mécaniques et ne comportent pas d'actionneurs. Elles sont peu coûteuses et légères. Seul le bras téléopéré MANUS a su trouver sa place parmi les utilisateurs de fauteuil roulant et son succès relatif est principalement dû au fait que la sécurité sociale hollandaise le rembourse. La preuve en est que la quasi totalité de ses utilisateurs vivent en Hollande et quasiment aucun de ses bras n'est utilisé en France.

Parmi les robots appartenant à la grande famille de la robotique de réhabilitation, les robots qui sont contrôlés par des médecins peuvent être distingués de ceux qui sont directement utilisés par des patients dans leur vie quotidienne, à domicile ou dans les centres de soins. Les premiers sont des outils évolués, réservés à un public d'expert. La question est alors de savoir quels outils robotiques sont utiles

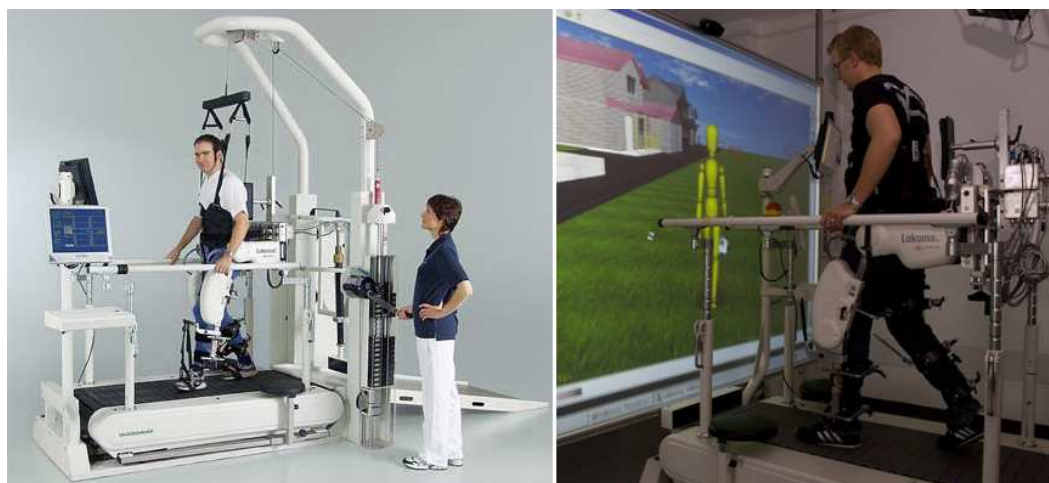


FIG. 1.4 – Le robot Lokomat rééduque la marche.

au corps médical et par voie de conséquence au patient. Dans le second cas, l'objectif est de palier à une déficience. Ces robots sont utilisés quotidiennement par des non spécialistes qui peuvent avoir des difficultés d'ordre moteur ou psychologique. Il s'agit alors de déterminer quelles aides techniques peuvent être utiles à des personnes en situation de handicap.

1.1.2 Les besoins des personnes en perte d'autonomie en termes d'assistance robotisée

Ce paragraphe présente les besoins exprimés par les personnes en situation de Handicap en termes d'assistance en général puis d'assistance technique. La première partie s'appuie sur une étude statistique réalisée auprès de la population française en 2004 et la suite se fonde sur les conclusions des validations d'outils robotiques réalisés par des roboticiens et des équipes médicales.

À la demande du Secrétariat d'état aux personnes handicapées et dans le cadre des débats parlementaires sur la réforme de la loi d'orientation en faveur des personnes handicapées de 1975, le CTNERHI (Centre Technique National d'Études et des Recherches sur les Handicaps et les Inadaptations), la DGAS (Direction Générale de l'Action Sociale) et la DREES (Direction de la Recherche des Études, de l'Évaluation et des Statistiques) ont rédigé une brochure intitulée "*Le handicap en chiffres*". Elle vise à proposer une photographie rapidement lisible des principaux chiffres clés sur le handicap et rappelle les définitions des principaux concepts et indicateurs [Brouard 04] (l'encadré 1.5 situe la notion de handicap).

Parmi la population française vivant à domicile³, 42% des personnes déclarent rencontrer des difficultés physiques liées à des déficiences⁴. 12% des déficiences déclarées sont attribuées à des accidents, 10% à des causes précoces (complication de grossesse, malformations congénitales, maladies héréditaires) et 26% au vieillissement. D'autre part, plus de 20% des personnes vivant à domicile déclarent au moins une incapacité⁵.

10% des personnes vivant à domicile déclarent avoir recours à une assistance régulière d'une autre personne. Inférieure à 7% avant 60 ans, cette proportion dépasse 20% après 70 ans. En 1999, deux

³Cette étude concerne la totalité de la population et non uniquement les personnes déclarées en situation de handicap.

⁴Déficience : correspond à un problème organique ou des structures anatomiques sous forme d'écart ou de perte importante. La notion de déficience ne comporte pas à proprement parlé de notion de gravité.

⁵Incapacité : difficulté ou impossibilité de réaliser des actes élémentaires comme se tenir debout, s'habiller, parler. Elles résultent en général, d'une ou plusieurs déficiences [Brouard 04].

millions d'adultes âgés de 20 à 59 ans et vivant à domicile avaient une pension d'invalidité ou un taux d'incapacité attribué par une instance administrative. 40% d'entre eux déclaraient recevoir une aide régulière en raison de leurs conditions de santé. Les aidants sont principalement des membres de la famille (62%) mais peuvent aussi être des aides professionnelles (13%). Au premier rang des tâches dispensées, on retrouve les tâches ménagères et les courses. En plus des aides humaines, près de 5.7 millions de personnes ont recours à au moins une aide technique⁶ à domicile. Les besoins en aide technique sont par ailleurs moins couverts à domicile qu'en institutions [Rousel 02]. Une analyse de l'apport des aides techniques mentionnée dans cette étude, réalisée dans [Nait Chabane 06] montre que si les besoins en aide à la mobilité sont presque satisfaits, les besoins en assistance à la manipulation ne le sont pas.

Il apparaît que nombre de personnes en situation de handicap pourraient être plus autonomes dans leur vie quotidienne si elles pouvaient regagner des degrés de mobilité ou bien saisir des objets en utilisant une aide technique adaptée. De nombreuses applications sont actuellement développées pour fournir une assistance robotique à cette population vieillissante. À tel point qu'une session spéciale *Human and Humanoid in Aging Society* a été dédiée à ce phénomène de société lors du congrès international de robotique (IEEE IROS) qui s'est tenu à Nice en 2008.

Cependant, s'il est possible d'établir les statistiques du handicap et de dresser la liste des fonctionnalités qui sont le plus atteintes par le vieillissement et le handicap, il est très difficile de traduire ces chiffres en cahier des charges d'une aide technique pour définir quelle aide robotisée serait la plus utile à une personne et sous quelle forme. Il est très difficile pour une personne handicapée d'imaginer à quoi ressemblera cette aide robotique *a priori*. Bien souvent, il y a un écart important entre ce que les personnes avaient demandé avant la réalisation d'un système et les remarques qu'elles font lors de phases de validation. Avant la conception du bras MANUS, une enquête auprès de personnes handicapées avait permis de déterminer qu'elles souhaitaient avoir un bras monté sur leur fauteuil, plutôt en avant du fauteuil et commandé par un joystick ou des boutons poussoirs. À l'usage, ces mêmes personnes ont constaté que ce bras alourdissait leur fauteuil et ne leur permettait pas de faire demi-tour dans un couloir étroit.

L'étude de Stranger *et al* [Stranger 94] tente de mettre en évidence à la fois les besoins prioritaires des personnes handicapées et les causes de la mauvaise distribution des outils de manipulation robotisés. Elle compile les résultats de neuf campagnes menées en Angleterre et aux États Unis qui dressent la liste des tâches prioritaires. Les questionnaires préliminaires, qui portent sur les capacités motrices et intellectuelles des utilisateurs, et les questionnaires post développement, qui portent sur l'évaluation de systèmes robotiques, sont conjointement étudiés. Globalement, ces travaux montrent que la possibilité de manipuler des objets du quotidien se trouvant dans un environnement inconnu est une priorité. Plus spécifiquement, la tâche qui arrive en tête de la liste des besoins prioritaires est le ramassage d'objets tombés au sol ou posés sur une étagère. Viennent ensuite les tâches relatives à la prise des repas, à l'hygiène et aux loisirs.

Une étude de 2005, réalisée par Balaguer [Balaguer 05], abonde également dans ce sens. Elle a sondé l'opinion de personnes handicapées à propos de l'utilisation d'une aide robotisée. 66% des personnes interrogées trouvent que la possibilité de saisir et relâcher des objets est primordiale. Environ 50% des personnes interrogées pensent qu'un bras robotique pourrait avoir un impact positif sur le niveau de soin quotidien dont ils ont besoin. Seuls 4.5% des sondés pensent qu'il aurait un effet négatif. D'autre part 86% des sujets pensent qu'ils pourront réaliser des nouvelles tâches dans leur vie quotidienne.

Il existe un besoin exprimé d'assistance à la saisie et à la manipulation d'objets et il a été montré que répondre à ce besoin permettrait à des personnes en perte d'autonomie de pouvoir rester plus longtemps à leur domicile, voire d'exercer une activité professionnelle. Pour que l'aide technique dé-

⁶Les aides techniques sont définies par la norme ISO9999 comme "tout produit, instrument, équipement ou système technique utilisé par une personne atteinte d'un handicap ou d'un désavantage social, fabriqué spécialement ou existant sur le marché, destiné à prévenir, compenser, soulager ou neutraliser la déficience, l'incapacité ou le handicap."

La notion de handicap

Apparu dans le langage usuel au milieu des années 1950, le terme handicap a été consacré en France, par la loi d'orientation en faveur des personnes handicapées de 1975. Cette loi abrogée en 2001, composante essentielle du système de protection sociale actuellement en vigueur, ne contenait ni définition du handicap, ni définition de la personne handicapée. Elle laissait aux COTOREP et CDES1 le soin de définir qui a accès aux hébergements, services et prestations en nature du secteur.

Le terme de handicap est vite apparu comme une notion recouvrant des concepts plus ou moins polymorphes selon les interlocuteurs, chacun insistant préférentiellement sur une des dimensions de ce polymorphisme : chronicité, incapacité, insertion sociale, nature de l'atteinte organique ou psychique. Pour tous, les contours du concept et, par voie de conséquence, ceux de la population concernée, étaient flous. Finalement, en France tout du moins, la population handicapée était soit celle, qui ayant déposé un dossier en CDES ou COTOREP était reconnue comme telle au travers de l'attribution d'une prestation spécifique du handicap, soit celle qui se considérait spontanément comme handicapée.

En 1980, l'Organisation Mondiale de la Santé a aidé à la clarification de la notion de handicap en adoptant à titre expérimental une Classification Internationale des Handicaps (CIH). Celle-ci a rapidement été considérée comme une avancée conceptuelle. Remplacée en 2001 par la Classification Internationale du Fonctionnement, du handicap et de la santé (CIF), la CIH constitue cependant encore la trame de l'analyse actuelle de la notion de handicap, notamment dans l'acceptation que ce concept est pluridimensionnel et que les différentes dimensions sont liées par des relations multiples.

Selon la CIF, le handicap, comme son corollaire positif, le fonctionnement, comporte deux grands types de composantes : la composante de l'organisme, elle-même constituée de deux domaines (celui des fonctions organiques et celui des structures anatomiques) et la composante des activités et de la participation (deux façons d'envisager les tâches et actions de la vie). La forme prise, pour un individu donné, par chacune de ces composantes est elle-même le résultat de l'interaction entre des problèmes de santé (trouble, maladie) d'une part, et des facteurs environnementaux et personnels d'autre part. Ainsi deux individus présentant le même problème de santé peuvent ils connaître des types ou des degrés de handicaps différents selon le milieu dans lequel ils vivent et leurs caractéristiques personnelles en termes d'âge, de sexe, d'origine sociale, d'expérience de vie préalable au trouble etc.

FIG. 1.5 – Extrait de la brochure *Le handicap en chiffres* [Brouard 04].

veloppée soit en adéquation avec les besoins exprimés, il est nécessaire d'inclure les équipes cliniques et les patients dans le cycle de développement des systèmes afin de faire évoluer le système vers un outil utile et accepté.

1.1.3 Quels critères doit satisfaire une aide robotisée destinée aux personnes en perte d'autonomie ?

Pour que des robots d'assistance soient diffusés chez les personnes en situation de Handicap, un compromis acceptable doit être trouvé entre les fonctions développées, la facilité d'utilisation, la fiabilité et le coût [Stranger 94]. Il faut également que le robot puisse fonctionner dans un milieu non structuré, qu'il soit le plus rapide possible et permette d'effectuer une grande variété de tâches [Kazi 95].

Une étude quantitative menée à la fin des années 80 [Batavia 91] a permis d'identifier 15 critères qu'un système d'assistance doit remplir en les classifiant par ordre de l'importance que les utilisateurs leur accordent. Cette liste a été affinée quelques années plus tard [Lane 96]. Il apparaît que le premier critère est l'*efficacité*, c'est-à-dire l'amélioration des conditions de vie de l'utilisateur, telle que perçue par l'utilisateur lui-même. Le second critère est l'*opérabilité*, il définit si le système est facile à commander et si sa commande est adaptée au handicap de la personne. Les critères recensés par l'étude de J.P. Lane sont donc les suivants [Lane 96] :

- | | |
|------------------|---|
| 1. Efficacité ; | 7. Sécurité et intégrité informatique ; |
| 2. Opérabilité ; | 8. Sécurité physique ; |
| 3. Coût ; | 9. Simplicité d'apprentissage ; |
| 4. Fiabilité ; | 10. Confort physique et acceptation ; |
| 5. Portabilité ; | 11. Facilité de maintenance. |
| 6. Durabilité ; | |

Nous allons revenir sur certains de ces critères qui ont plus particulièrement guidé les choix de développement de notre projet.

1.1.3.1 Efficacité, fiabilité et robustesse

Pour que l'utilisateur ait confiance dans un système qui fera partie intégrante de sa vie quotidienne et accepte de l'utiliser, la fiabilité et l'efficacité sont des enjeux majeurs. Un utilisateur préférera être sollicité voire même téléopérer le système, si ses facultés motrices le permettent, plutôt que de voir une tâche automatique échouer. D'autre part, pour qu'un système de saisie soit utilisable quotidiennement, il doit permettre de saisir le plus vaste choix d'objets possible avec un minimum de restriction sur les objets [Lane 96].

Ainsi, lors du développement des outils présentés dans cette étude, la possibilité de saisir un objet quelconque sera privilégiée sur le temps mis pour effectuer la tâche de saisie. Lors du forum dédié aux attentes des personnes handicapées organisé lors de la Conférence Internationale sur la Robotique de Réhabilitation (ICORR'07), il est apparu qu'une saisie autonome doit pouvoir être effectuée dans un temps qui a le même ordre de grandeur que le temps que mettrait une saisie en mode manuel.

D'autre part, l'utilisateur peut être sollicité dans la mesure de ses facultés pour contrôler le système et le rendre plus robuste. La question est alors de savoir comment partager au mieux les tâches entre le système et la personne pour assurer la réalisation d'une tâche tout en limitant la fatigue de l'utilisateur.

1.1.3.2 L'opérabilité ou l'interface de commande

Les systèmes dédiés aux personnes en perte d'autonomie doivent disposer d'une interface adaptable à la variété des handicaps représentés. Le chapitre du *handbook of robotics* [VanderLoos 08] souligne trois différences majeures entre les robots industriels et les robots d'assistance. Tout d'abord, l'opérateur n'est pas un spécialiste en commande robotique, ce n'est ni un roboticien ni un technicien.

Ensuite, l'opérateur présente par définition un handicap. Enfin, l'interface doit être individualisée : une solution qui fonctionne pour un patient ne fonctionnera pas telle quelle pour un autre patient, ne serait-ce que pour des questions de morphologie. Le système doit pouvoir être modifié par un ergothérapeute pour être adapté au patient auquel il est prescrit.

Un soin particulier doit être porté aux interfaces de contrôle qui doivent être intuitives pour que le plus grand nombre puisse les utiliser. Plus son handicap affecte une personne et plus chaque action est coûteuse et pénible. L'idéal est de proposer à l'opérateur d'utiliser une interface à laquelle il est habitué comme le proposent les ARPH [Colle 02] et les projets AVISO et ANSO du CEA-LIST [Remazeilles 08, Leroux 06].

Notre étude fait suite au projet AVISO qui propose une interface graphique permettant de contrôler le système à partir de *clics* sur un écran en utilisant l'outil le plus adapté au handicap de la personne (track ball, déclenchement au souffle, suivi du regard, Joystick...). Il a été montré que plus le nombre de validations (ou *clics*) de l'utilisateur est élevé, plus la fatigue est importante [Leroux 06]. Ainsi, l'objectif de cette étude sera de réduire la commande à un *clac* unique sur une interface graphique, généré par le périphérique le plus adapté à l'utilisateur.

1.1.3.3 Sécurité

Lorsqu'un robot d'assistance est déployé à proximité des personnes, la question de la sécurité devient fondamentale. En effet, la robotique d'assistance ne permet pas d'isoler les robots. Il faut donc trouver des solutions pour rendre les systèmes inoffensifs. C'est actuellement l'un des freins principaux au développement de systèmes de manipulation pour le handicap.

Pour sécuriser un robot d'assistance, des solutions peuvent être trouvées dans sa conception mécanique. Par exemple, pour limiter le couple en bout de bras, la société Exact Dynamics a placé tous les actionneurs du MANUS dans sa base. Ainsi, le bras est allégé du poids des moteurs et le couple en bout de bras est réduit ce qui minimise la force d'un impact éventuel. Il est également possible d'ajouter des capteurs d'effort sur le bras afin d'arrêter son mouvement en cas de contact non désiré.

Avant de pouvoir être testé en milieu hospitalier ou lancé sur le marché, un système robotique doit passer une série de tests pour recevoir une habilitation auprès du Comité de Protection des Personnes (CPP). Les tests réalisés assurent que les défaillances élémentaires ne conduisent pas à un mode de fonctionnement dangereux pour la personne. Le bras MANUS possède les habilitations pour être utilisé à proximité des patients. Cependant, à chaque modification du système il faut demander et obtenir à nouveau ces habilitations.

Si la sécurisation mécanique du bras peut être quantifiée et vérifiée, il existe une autre notion de sécurité bien plus subjective et difficilement quantifiable par les normes : l'appréciation de l'utilisateur de sa propre sécurité. Les personnes qui utilisent un bras sur fauteuil sont, par exemple, souvent anxieuses de voir le bras s'approcher trop près de leurs jambes. Elles craignent d'être blessées sans sentir la douleur. On se rapproche alors de la notion de fiabilité. Plus le système est robuste et fiable et plus cette sensation de sécurité sera améliorée.

Pour palier à cette sensation d'insécurité, l'utilisateur doit pouvoir reprendre le contrôle sur le robot à tout moment pour l'arrêter ou modifier son comportement. Il faut qu'il ait en permanence une possibilité de contrôle souveraine par rapport au contrôle autonome du robot.

1.1.3.4 Coût

Si les systèmes médicaux classiques sont principalement financés par des instituts, les systèmes d'assistance, quant à eux, sont principalement financés par les particuliers [Stout 08]. En France, il peuvent être en partie pris en charge par des systèmes de financement de la sécurité sociale (via les MDPH). Le problème du coût est étroitement lié à la personnalisation nécessaire du matériel pour qu'il soit adapté au handicap de son utilisateur. Pour donner un exemple, un fauteuil roulant

électrique contrôlé par des périphériques adaptés vaut le prix d'une voiture de classe moyenne alors qu'elle embarque nettement moins de technologie.

Contrairement au secteur automobile, le secteur du handicap ne peut pas compter sur une production de masse pour faire diminuer les coûts. Le chapitre [VanderLoos 08] dédié à la robotique de réhabilitation du *Handbook of robotics* propose de résoudre ce problème en recourant à la *personnalisation de masse*. Plutôt que de parler de robot d'assistance aux personnes handicapées, pourquoi ne pas étendre l'assistance à un public plus large, comme celui des *papis boomers* qui ont des moyens financiers, en rebaptisant les robots d'assistance en *robots d'amélioration du quotidien*. La base du marché est élargie et une production suffisante à la baisse du prix peut alors être envisagée.

Dans cette étude, nous nous attacherons à limiter le prix de l'outil de saisie en choisissant des capteurs standards et peu coûteux.

1.1.3.5 Faible encombrement

Il existe plusieurs manières d'envisager l'aide au handicap. La première solution consiste à aménager l'environnement pour qu'il soit adapté au handicap de la personne qui y évolue. Cette solution peut être envisagée lors de la conception de logements ou de centres de soins. Pour une personne en fauteuil, il s'agit par exemple d'élargir les espaces de circulation, d'installer des portes coulissantes, des rampes d'accès et des ascenseurs et d'étudier le mobilier pour que les objets soient accessibles. À Lorient, le centre de rééducation et de réadaptation fonctionnelle de Kerpape teste des appartements témoins de ce type.

Si un environnement adapté semble être la solution idéale, elle ne peut pas être entièrement étendue à l'espace public où personnes en situation de handicap et personnes valides se côtoient. En effet, si de nombreux efforts sont faits pour l'accessibilité (places de parking, rampes et ascenseurs) très peu d'efforts sont faits pour permettre à des personnes handicapées, par exemple, d'atteindre des objets sur les étagères marchands. D'autre part, il est difficile dans certaines villes d'acquiescer un logement suffisamment spacieux pour mettre en place les aménagements nécessaires. Dès lors, plutôt que d'adapter l'environnement au handicap de la personne, il est préférable d'adapter, dans la mesure du possible, les outils pour le handicap aux espaces publics.

L'encombrement du système devient alors un point clé.

Conclusion

La saisie et la manipulation d'objets apparaissent comme des tâches prioritaires pour les personnes âgées et handicapées. Si elles disposaient d'un outil de saisie d'objets, elles pourraient regagner une certaine autonomie qui leur permettrait de se passer de l'aide d'un assistant quelques heures par jours pour manger, lire un livre, prendre un objet sur une étagère ou le ranger, faire leurs courses. En plus de leur permettre de vivre chez elles plus longtemps, l'assistance à la saisie pourrait aussi leur permettre de retrouver une place dans la vie active.

S'il est aisé de recenser les besoins des personnes handicapées, il est très difficile de traduire ces besoins en termes d'aide technique. En effet, il est impossible pour une personne de déterminer à l'avance quelle devrait être la structure du robot dont elle aurait besoin. La seule solution est d'inclure équipes médicales et patients dans le cycle de développement des systèmes pour avoir un retour sur les fonctionnalités développées au fur et à mesure de leur développement.

Pour qu'un robot d'assistance trouve sa place auprès des personnes en perte d'autonomie, il doit réunir un certain nombre de critères qui ont été établis par des études statistiques auprès de la population concernée. Ce système doit en particulier être efficace et facilement opérable. Viennent ensuite des impératifs de coût, de sécurité et d'encombrement.

La section suivante présente les assistants à la saisie existants. En particulier, seront étudiées les hypothèses émises sur les objets et les scènes où ils se trouvent, que ces hypothèses soient directement

imposées par la structure du robot ou bien utilisées pour simplifier le problème de la saisie. Les méthodes de désignation des objets et les modes de commande seront également détaillés.

1.2 Les systèmes d'aide à la saisie

La saisie d'objet en *milieu humain* est un challenge technique qu'il faudra résoudre pour développer des robots qui puissent assister les personnes en perte d'autonomie dans leur vie quotidienne [Kemp 07, Katz 08]. Plusieurs caractéristiques de l'environnement humain ont été identifiées et décrites dans [Kemp 07]. L'environnement humain est un environnement non contrôlé et, par définition, dimensionné pour les humains, dynamique (les objets changent de place sans que le robot agisse). La difficulté principale est de percevoir cet environnement, à tel point que la plupart des aides techniques à la saisie en milieu humain sont téléopérées ou limitées à une utilisation dans un *environnement structuré*, c'est-à-dire modifié pour être adapté à l'utilisation du système, typiquement l'environnement d'un robot industriel dans une usine. Dans la suite de ce document, nous opposerons les *environnements humains* aux *environnements structurés*.

Les outils robotiques d'assistance à la saisie existants sont généralement différenciés en plusieurs catégories selon qu'ils sont fixes ou mobiles, dépourvus ou équipés de capteurs [Kawamura 94a, Kawamura 94b]. Cette section n'a pas pour vocation de dresser un état de l'art exhaustif des systèmes existants mais plutôt d'observer quelles sont les contraintes liées à chacune de ces catégories physiques. Le lecteur intéressé pourra se référer à l'état de l'art des systèmes robotiques pour le handicap du chapitre dédié du *Handbook of robotics* [VanderLoos 08]. Une perspective historique des systèmes pour le handicap en Europe peut être également trouvée dans [Hillman 03, Dallaway 95]. Le pendant américain de cette étude est [Harwin 95].

Dans un premier temps, les systèmes de saisie d'objets sans capteurs seront présentés. Ils permettent de saisir des objets dont l'emplacement et la position sont parfaitement déterminés en *mode automatique* préprogrammé. Ils permettent également de saisir des objets en *mode manuel*, c'est l'utilisateur qui perçoit l'environnement et adapte les mouvements du système à la tâche à effectuer. Le deuxième paragraphe est dédié aux systèmes avec capteurs qui imposent des contraintes permettant de limiter l'environnement ou l'ensemble des objets. Enfin, nous nous intéresserons aux systèmes qui proposent des méthodes de saisie d'objets dans des environnements ouverts pour pouvoir saisir la plus grande variété d'objets dans un milieu non structuré.

1.2.1 La saisie d'objet sans perception de l'environnement.

Les stations de travail constituent le premier transfert de la robotique industrielle vers la robotique d'assistance dans les années 60 [Hillman 03]. Elles étaient destinées aux activités de la vie courante ou professionnelle telles que les activités de bureau, la préparation des repas ou encore l'assistance à la saisie d'objets à proximité d'un lit. Elles sont composées d'un ensemble de systèmes électromécaniques installés dans une pièce dédiée à leur utilisation. La plupart du temps, ces plateformes ne sont pas équipées de capteurs.

Le projet européen RAID, pour *Robotic Assistance in Daily living*, permet à une personne handicapée d'exercer une activité professionnelle de bureau [Jones 99, Busnel 01]. Le robot RAID est constitué d'un bras industriel monté sur un ensemble bureau-étagère (voir la figure 1.6). Chaque alvéole de l'étagère contient des documents répertoriés dont l'emplacement est connu du système. Son interface de commande est basée sur les développements du projet MASTER [Busnel 01] qui permet à une personne handicapée d'utiliser le périphérique de commande le plus adapté à son handicap pour contrôler les mouvements du robot : contrôle vocal, joystick, smart ball, etc.

Le projet collaboratif américain DEVAR, *Desktop Vocational Assistant Robot*, de l'université de

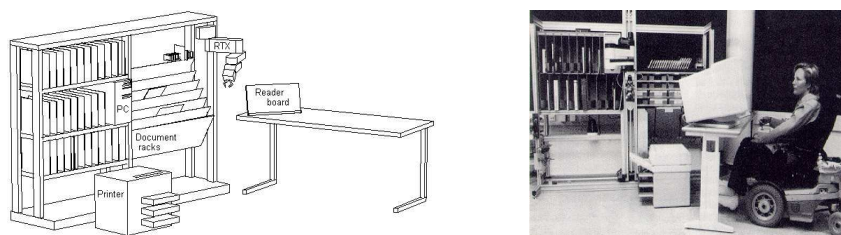


FIG. 1.6 – La station de travail Raid (robotic assistance in daily Living) [Jones 99]

Stanford et du centre médical VA (1989-1994) [VanderLoos 95], est un autre exemple de station de travail. Il a été poursuivi et renommé par la suite PROVAR [VanderLoos 00] (1996-1999). Cette fois, le bras manipulateur est fixé sur un système de rails au plafond (voir la figure 1.7). Il est équipé de capteurs ultra-sons qui permettent de détecter et d'éviter des obstacles lors de la manipulation dans des modes de commande téléopérés. L'environnement est contraint. Le système connaît l'emplacement de chaque objet et de chaque rangement. Le bras est programmé pour atteindre des emplacements précis sur le bureau ou sur des étagères pour y chercher des objets spécifiques (du papier, un téléphone, un livre) dont la position est parfaitement connue. Le positionnement de ces objets aux emplacements prévus demande l'intervention quotidienne d'un assistant valide. Ces stations de travail sont donc limitées par leur manque de réactivité face à un changement de leur environnement : s'il y a la moindre erreur entre la position courante d'un objet et la position apprise par le système, il lui est impossible de saisir cet objet.

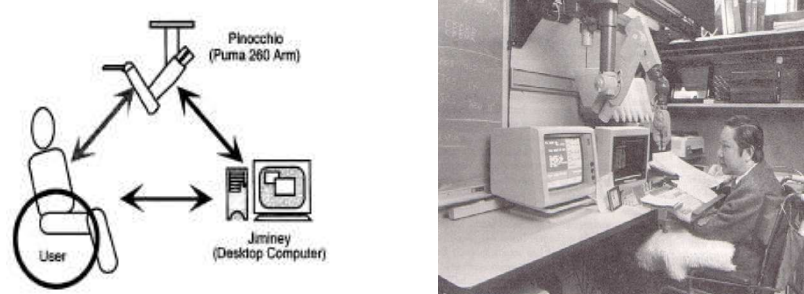


FIG. 1.7 – Le projet DeVAR/ProVAR [VanderLoos 95, VanderLoos 00]

Cependant, cette stratégie de saisie est simple. Elle a été directement portée sur des petits systèmes légers que l'on pose sur une table pour manger, se laver ou tourner les pages d'un livre. Ces robots, vendus comme du petit matériel électroménager, sont les premiers véritables succès commerciaux en terme de robotique d'assistance. Ce sont des systèmes électromécaniques simples et peu coûteux qui sont conçus pour réaliser des tâches spécifiques.

Le robot Handy I [Topping 98] est un bras monté sur table roulante qui présente un plateau à compartiments (voir la figure 1.8). Le plateau et l'instrument en bout de bras peuvent être changés selon la tâche à effectuer (manger, se maquiller, se laver).

MySpoon [Ishii 95, Soyama 03] de la société Secom est un petit système qui se pose sur une table pour l'aide à la prise des repas. Il est également constitué d'un plateau compartimenté et d'un bras qui est commandé par un joystick. Pour utiliser ces deux systèmes, l'intervention d'un assistant valide est nécessaire pour mettre en place les accessoires ou placer les aliments préalablement découpés dans les compartiments.

Pour se libérer de la contrainte de positionnement des objets imposée sur les robots qui viennent d'être présentés, une solution est donnée par les modes de commande manuelle. Un des premiers



FIG. 1.8 – *Le robot Handy*



FIG. 1.9 – *Le robot my spoon*

bras fixes pour l'assistance à la saisie d'objets placés dans une position quelconque, est le robot Tou [Casals 93] développé à l'université Polytechnique de Catalogne en Espagne (voir figure 1.10). Il a été conçu initialement pour pouvoir saisir des objets ou refaire un lit. Ce bras ne dispose d'aucun capteur, il ne fonctionne qu'en mode manuel. L'utilisateur commande TOU en utilisant la voix ou sa mobilité résiduelle (joystick ou clavier spécial).

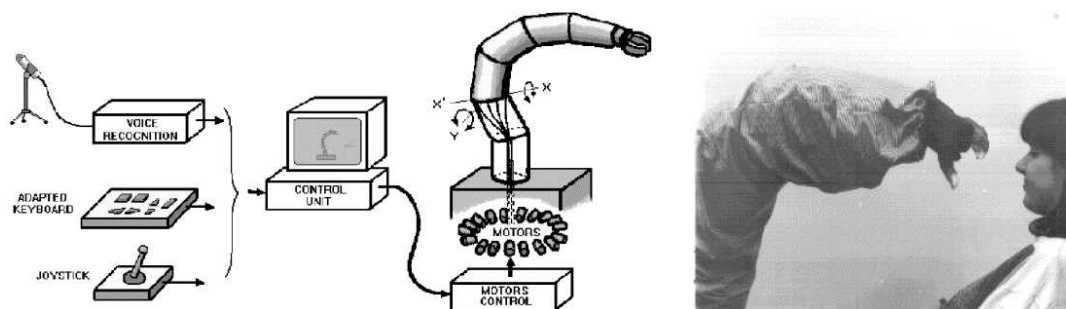


FIG. 1.10 – Le robot Tou

Pour augmenter l'espace d'utilisation de l'outil de saisie, des bras légers et portables se fixant sur les fauteuils roulants ont été mis au point dans les années 80. Les premiers bras pour fauteuil roulant commercialisés sont le **Manus Arm** [Eftring 99, Kwee 89] d'Exact Dynamics (6 + 2 degrés de liberté) et le bras Raptor de Phybotics [Mahoney 01] (4 degrés de liberté). Leurs effecteurs sont des pinces parallèles à deux doigts. Ce sont des bras sécurisés qui sont faits pour être déployés à proximité des utilisateurs. Ces systèmes sont téléopérés via un jeu de commandes standards. Le bras Manus est livré avec un clavier 16 touches, des pédales, et un joystick. Il présente deux modes de commande, une commande cartésienne qui permet de déplacer la pince selon les trois directions principales d'un repère lié à la base du bras ; et un mode de commande articulaire qui permet de déplacer chacune des articulations du bras indépendamment les unes des autres.

Ces deux bras ont été sujets d'une étude mécanique comparative [Sundarrao 06] montrant que les six degrés de liberté du bras MANUS permettent une bonne manipulabilité tandis que les 4 degrés du Raptor sont tout juste suffisants pour réaliser une tâche de saisie. Cette étude met aussi en évidence les difficultés d'utilisation des commandes standards selon le handicap de la personnes et les défauts des contrôleurs natifs de ces bras. Ces bras ont été conçus pour être téléopérés. Le contrôleur natif du Manus est particulièrement inadapté à une commande autonome via un ordinateur : la précision de la commande en vitesse est très faible ($5^\circ/s$) et les temps de réponse sont très importants.

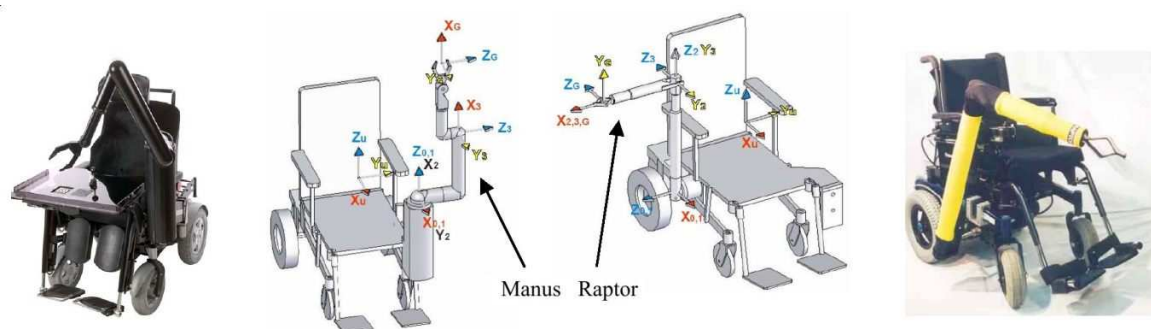


FIG. 1.11 – Les bras Manus et Raptor

Les possesseurs d'un bras Manus arrivent à manipuler et à saisir de nombreux objets et même à déplacer leurs membres (voir figure 1.2.1). Cependant certaines situations peuvent être inextricables

dans un mode manuel, typiquement lorsque l'objet à saisir n'est pas dans le champ de vision de la personne. Il est particulièrement difficile de ramasser un objet au sol ou placé sur une étagère en hauteur. D'autre part, la commande de ces bras nécessite une certaine mobilité ce qui les rend inaccessibles à certains types de handicap.



FIG. 1.12 – Le bras Manus est un bras sécurisé qui peut être commandé en utilisant un joystick

Récemment, le projet espagnol ASIBOT a proposé une solution de fixation polyvalente pour le bras manipulateur Mats (voir 1.13). Ce bras peut être fixé à un fauteuil roulant, il peut être branché à une prise fixée à une table ou bien encore se déplacer de pièce en pièce en se branchant sur un réseau de prises fixées aux murs. Ce bras ne dispose pas de capteurs lui permettant de recueillir des informations sur son environnement. Il est entièrement téléopéré. Il peut se déplacer automatiquement d'une station de travail à l'autre et être emmené à l'extérieur.



FIG. 1.13 – Le robot Mats peut se déplacer de prise en prise à l'intérieur d'une maison, se fixer sur un support fixe ou encore sur un fauteuil roulant

Comme aucun capteur n'équipe ces robots, ils n'ont aucune perception de leur environnement. Il n'y a alors que deux solutions pour permettre la saisie d'objets : soit l'environnement est entièrement contraint, soit l'utilisateur commande manuellement tous les mouvements du robot.

À court terme, concevoir des outils de saisie sans capteur est un moyen efficace pour parvenir à une commercialisation rapide des produits [Gates 07], en témoignent la diffusion importante des outils MySpoon, Handy et le nombre de bras Manus utilisés. Cependant, ces outils ne permettent pas de répondre au problème de la saisie et de la manipulation d'objets du quotidien pour tous les types de handicap. En effet, l'utilisation des bras en mode téléopéré demande une certaine dextérité et beaucoup d'attention, qui ne peuvent être garanties selon le degrés du handicap de l'opérateur. Par exemple, le champ d'utilisateurs potentiels du bras MANUS ou de MySpoon est restreint aux personnes qui peuvent se servir d'un joystick ou d'un pavé numérique. De plus, les outils MySpoon et Handy nécessitent l'intervention d'une personne valide pour préparer le système.

1.2.2 La saisie d'objets en environnement limité

Afin de réduire la sollicitation de l'utilisateur, des projets ultérieurs ont équipé les outils de saisie de capteurs, pour proposer des modes de commande partagés, voire autonomes. Les premiers systèmes équipés de capteurs sont des manipulateurs fixes qui permettent de saisir des objets qui sont placés sur un plan de travail uni et de dimensions connues. La saisie des objets est donc spatialement limitée et le fond uni permet de repérer les objets très simplement en utilisant une caméra. Ces conditions sont similaires aux conditions contrôlées rencontrées dans le contexte industriel. Il est donc possible de transférer les outils développés pour l'industrie vers ce type d'applications. Ces environnements dont on connaît parfaitement les caractéristiques sont dits *structurés*.

Environnement limité à une zone de l'espace Le robot ISAC (Intelligent Soft Arm Control) [Kawamura 94a] et le robot CORA (COoperative Robot Assistant) [Iossifidis 02] sont deux exemples de systèmes de saisie qui fonctionnent en environnement structuré (voir les figures 1.14 et 1.15). ISAC est un outil d'aide à la prise des repas et CORA est un outil de saisie. Ces deux systèmes sont des plateformes fixes équipées de caméras qui permettent d'observer la scène et l'utilisateur. Les objets à saisir sont placés sur un support uniforme et sont désignés par la voix ou bien montrés du doigt. Dans ce cadre, les méthodes développées pour détecter les objets et le visage ou les mains de l'utilisateur tirent parti des avantages d'un environnement fixe, en particulier, une phase de calibration permet de détecter la couleur de la peau pour la lumière courante et pour un utilisateur donné facilitant ainsi la détection des mains et du visage.

Dans les deux cas, le scénario est identique, l'utilisateur désigne l'objet à saisir par une commande vocale, le robot identifie cet objet parmi les objets détectés sur le support et déclenche la saisie. La reconnaissance des objets suppose l'existence d'une base de données construite au préalable, ce qui a pour effet de limiter la saisie à un ensemble d'objets dont les caractéristiques visuelles et les positions de saisie sont enregistrées. Dans le cas où la reconnaissance échoue, le scénario de CORA propose à l'utilisateur de montrer l'objet du doigt. De la position de la main est déduite la direction de cet objet qui est ensuite détecté puis saisi en utilisant uniquement des caractéristiques géométriques calculées en ligne. Notons que l'objet à saisir est un tournevis et que cet objet est simple à saisir.

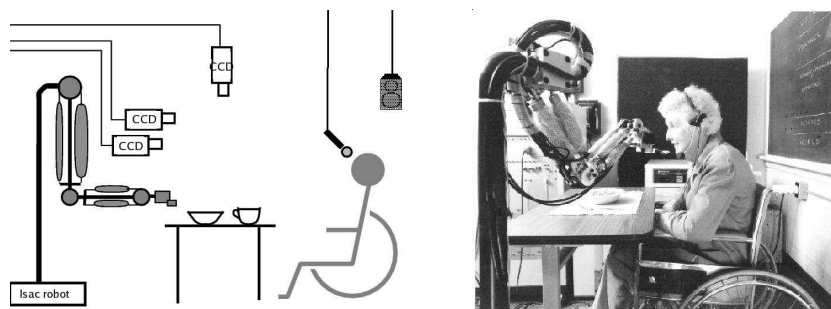


FIG. 1.14 – Le robot fixe ISAC

Si la reconnaissance de l'objet échoue ou bien si l'objet à saisir est observé pour la première fois, seules les informations obtenues en ligne peuvent être utilisées pour permettre la saisie. Cette information peut être utilisée pour calculer les points de saisie pour des objets plans [Sanz 05] ou des volumes [Speth 08].

Si l'on considère maintenant des robots mobiles, il n'est pas possible de contraindre leur voisinage, il évolue donc dans un milieu non structuré qui demande de développer de nouvelles méthodes pour saisir des objets dans une grande variété de scènes.

Une première solution consiste à se placer dans des conditions qui permettent de revenir à un environnement structuré, par exemple en limitant la zone de saisie à une table connue ou à un plateau porté par le système. Le robot Friend est un bras MANUS fixé sur fauteuil équipé de caméras et d'un

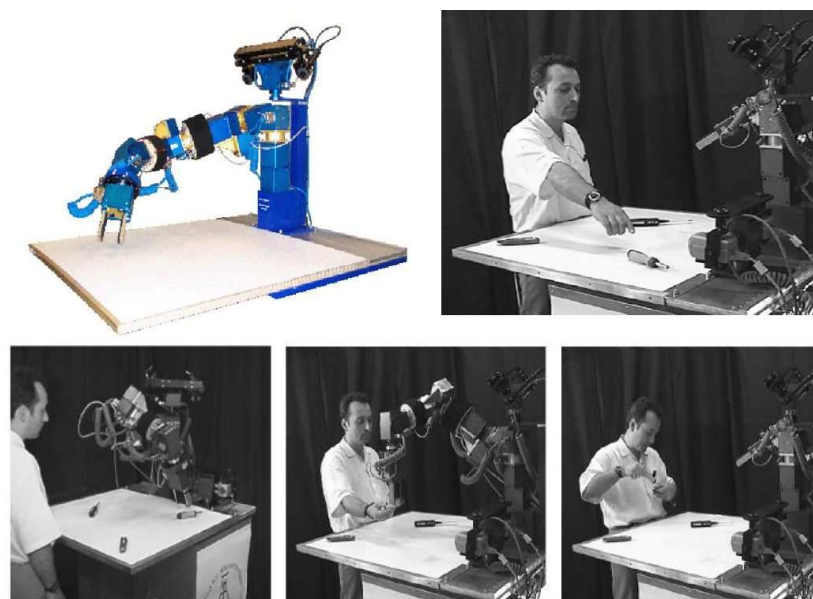


FIG. 1.15 – La station fixe CORA

plateau intelligent. Il s'agit en fait d'un grand capteur tactile qui repère la position et la forme des objets placés à sa surface. En croisant l'information obtenue par le plateau et celle donnée par les caméras [Volosyak 05b, Volosyak 05a] l'objet peut être localisé (voir la figure (1.16)). Cependant, si l'utilisation du plateau simplifie la détection, elle requiert une étape préalable de placement des objets.

Environnement limité à un nombre d'objets restreint. Dès lors que le système est monté sur un support mobile, la zone de saisie devient potentiellement illimitée et les outils de saisie doivent alors pouvoir être utilisés pour une grande variété de scènes : le support des objets n'est pas nécessairement plan et peut être texturé, l'illumination ne peut pas être contrôlée. Pour saisir des objets, une solution est alors d'imposer des contraintes sur les objets eux-mêmes. L'environnement n'est pas limité à une portion de l'espace mais à un nombre restreint d'objets.

Seuls les objets dont certaines caractéristiques sont enregistrées dans une base de données peuvent être saisis. Ces caractéristiques peuvent être de plusieurs types :

- marqueurs artificiels fixés aux objets, puce RFID ou marqueurs visuels ;
- indices visuels repérés dans des vues de l'objet ;
- modèle 3D de l'objet.

Les informations relatives à chaque objet peuvent être complétées par sa dénomination, les positions potentielles de saisie, des scénarios d'utilisation, des liens vers les objets auxquels il est généralement associé, etc. Ainsi, il est possible de désigner l'objet par la voix ou en désignant une vue le représentant sur un écran affichant une liste d'objets. Une fois l'objet détecté, il peut être précisément localisé à partir des informations disponibles dans la base de données.

La première version du projet FRIEND et le robot KARES (KAIST Réhabilitation Engineering System) [Song 98] sont des bras MANUS montés sur fauteuil et équipés d'un système de vision. Les objets à saisir sont marqués par des autocollants dont la couleur, la forme et la position sur l'objet sont connues. La saisie s'effectue en alignant une caméra montée sur le bras avec les marques portées par les objets. Le traitement de l'image est simplifié puisqu'il s'agit de repérer des marqueurs artificiels conçus pour être facilement identifiables.

La figure (1.17) présente l'assistant CARE-O-BOT [Graf 04], un manipulateur mobile qui est à la

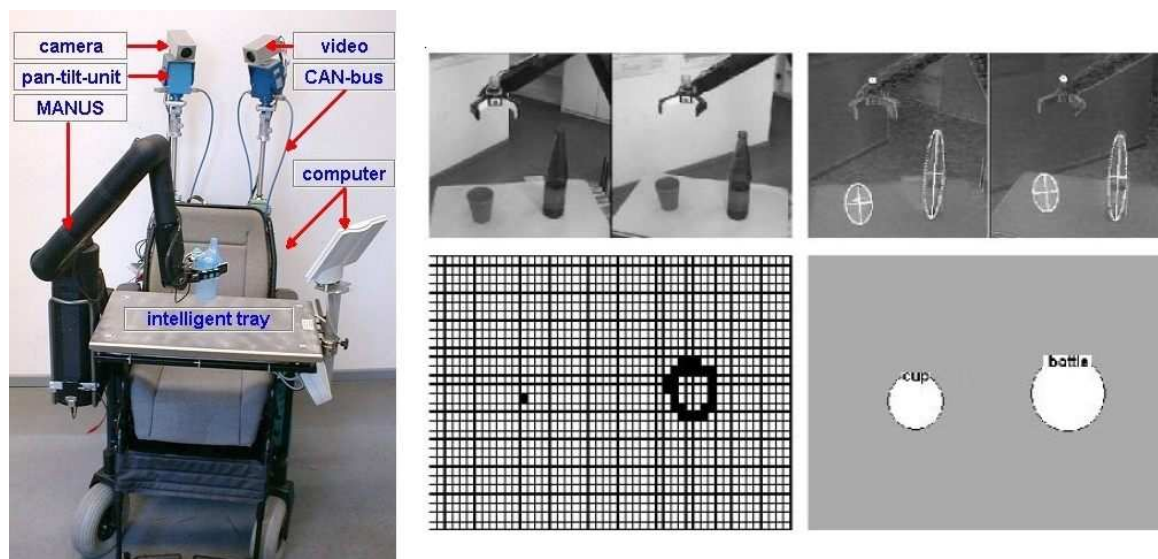


FIG. 1.16 – Le robot Friend et son système de localisation d'objet basé sur la vision et un plateau tactile

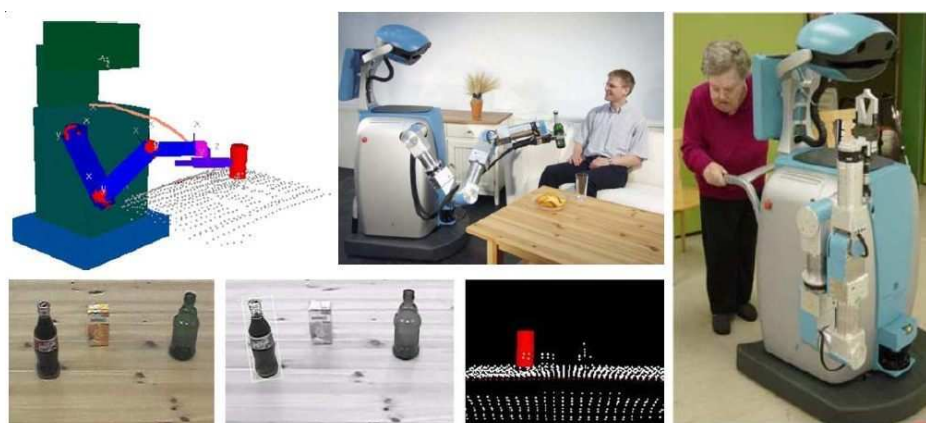


FIG. 1.17 – Le projet allemand CARE-O-BOT utilise à la fois un laser et une caméra pour détecter les objets et estimer leur forme

fois une aide à la saisie et une aide à la marche. Le système est équipé de caméras et de télémètres laser. Il dispose d'une base de données d'apparences d'objets qui lui permet d'identifier dans la vue courante les objets présents s'ils ont été enregistrés auparavant. Une fois les différents objets composants la scène identifiés, l'utilisateur désigne l'objet qu'il souhaite saisir par la voix et le robot focalise son système de vision et son télémètre laser sur l'objet. La position de l'objet est d'abord évaluée par mise en correspondance des informations visuelles courantes et stockées dans la base de données, puis affinée par les données fournies par le télémètre laser.

Plutôt que de stocker les indices sous la forme d'une liste de données 2D, la base peut être établie autour d'une représentation 3D de l'objet [Liefhebber 07]. En plus d'informations sur le voisinage photométrique des points, une information sur leurs positions relatives est stockée. Le projet TNO propose d'extraire des descripteurs visuels dans plusieurs vues d'un objet et de les mettre en correspondance pour estimer leur position 3D. Cette méthode permet d'obtenir une estimation de la pose de l'objet dans le repère caméra alors que les méthodes basées sur une liste d'informations 2D permettent d'estimer l'homographie entre la vue courante et la vue de la base de données qui lui correspond.

Les robots Robonaut [Peters 06] et HRP-2 [Sian 07] (voir figure (1.18)) disposent d'une base de données de modèles CAO des objets et des meubles et tentent de retrouver ces modèles dans l'environnement par détection de contours. Le robot Robonaut apprend le comportement à adopter en enregistrant des phases téléopérées. Le robot humanoïde HRP-2 est équipé d'un système de vision qui lui permet de détecter un réfrigérateur, de l'ouvrir et de détecter une canette de soda à l'intérieur.



FIG. 1.18 – Le robot HRP-2 dispose d'une base de données des modèles CAO de son environnement. Le scénario consiste à détecter un réfrigérateur, à l'ouvrir et à saisir une bouteille de soda à l'intérieur. Le système connaît a priori les modèles de la canette et du réfrigérateur.

L'utilisation de bases de données d'objets marqués, d'apparences, ou de modèles permet de réaliser des saisies dans des scènes quelconques en utilisant des informations visuelles. En particulier, lorsque des marqueurs sont utilisés, le traitement de l'information visuel est trivial et efficace. Si cette solution est pertinente pour une utilisation au domicile d'une personne où un même objet est fréquemment rencontré, le marquage des objets limite l'opérabilité de l'outil : les techniques de reconnaissance d'objets peuvent échouer en milieu ouvert et ne sont pas adaptées à la saisie d'un objet non répertorié. La question est alors de savoir s'il est possible d'utiliser les informations contenues dans une base de données contruite à partir d'une liste d'objets restreinte pour saisir un objet qui n'appartient pas à cette liste. Plusieurs travaux récents proposent des méthodes pour saisir un objet inconnu à partir de connaissances *a priori*, par exemple en créant non pas une base d'apparence des objets mais une base d'apparence des points de saisie [Saxena 08], ou encore en créant une base de modèles génériques [Curtis 08].

Une autre manière d'aborder ce problème est d'inclure l'utilisateur dans la détermination de la stratégie de saisie. S'il n'est pas trivial, pour un robot, de mettre en correspondance les informations stockées avec les informations acquises en ligne, l'identification de l'objet courant est immédiate pour l'utilisateur. C'est de ce constat que part la solution apportée par le projet MUSIC [Kazi 95] qui est un assistant à la saisie d'objet à partir d'informations données par des capteurs de modalités multiples (micro, caméra, laser, etc.). La saisie repose sur une hiérarchie de données dont le sommet est un

modèle générique d'objets qui est de plus en plus spécifié au fil des étages hiérarchiques. Afin de ne pas se confronter au problème de la reconnaissance d'objet automatique, c'est l'utilisateur qui pointe à l'aide d'un laser puis identifie l'objet et en communique la nature au système par l'intermédiaire d'une reconnaissance vocale.

Outre la reconnaissance des objets à partir d'informations obtenues en ligne, l'utilisation de bases de données contruites hors ligne pose d'autres problèmes : est-il possible d'estimer la pose de l'objet de manière suffisamment précise pour pouvoir positionner les mors de la pince exactement dans la configuration enregistrée ? Que se passe-t-il si un obstacle se trouve à la position que la pince devrait occuper ? Comment traiter le cas d'un objet non reconnu à cause d'une mauvaise perception ou bien parce qu'il est rencontré pour la première fois ?

1.2.3 La saisie d'objets en environnement humain

Le credo émergeant pour saisir des objets en milieu humain est qu'il n'est pas nécessaire de connaître un objet dans ses moindres détails pour pouvoir le saisir. Actuellement, les études sur la saisie d'objets tentent de se libérer de l'utilisation de bases de données exhaustives. La plupart des objets du quotidien ont des structures communes qui sont destinées à simplifier leur utilisation [Norman 90]. En concevant des saisies qui s'adaptent à ces structures, on simplifie la manipulation.

Certains scénarios de saisie reposent sur des bases de données de catégories suffisamment discriminantes pour adapter la saisie à la forme de l'objet [Taylor 02, Saxena 08]. Saxena de Stanford a proposé un travail très prometteur sur la détection de points de saisie pour tous les objets du quotidien en utilisant une seule caméra. Le détecteur est entraîné en utilisant les modèles 3D simulés de cinq objets sur lesquels les points de saisie sont marqués : un livre, une tasse, un crayon, une boîte et un verre à cocktail (voir figure Saxena 1.19). En utilisant ce détecteur de points de saisie, le robot STAIR, un manipulateur mobile équipé d'un système de vision, parvient à saisir un grand nombre d'objets qui ne sont pas identiques à ceux qui ont servis à constituer la base de données. L'algorithme a été testé pour des scènes simples qui contiennent un objet unique sur un fond uniforme dont la couleur contraste avec la couleur de l'objet. Il n'est pas certain que la méthode puisse être étendue à tout type de scène. Cependant cette approche est très prometteuse car elle montre que des données obtenues en simulation peuvent être utilisées pour classifier des données réelles. Cela permet de simplifier l'élaboration de bases de données.

Taylor et Kleeman [Taylor 02] proposent d'utiliser un rayon laser rouge et une caméra. Le rayon laser est détecté par segmentation couleur dans l'image de la caméra et sa déformation est utilisée pour reconstruire les objets composant la scène observée. Les formes obtenues sont ensuite classifiées en plusieurs catégories. Une stratégie de saisie est associée à chacune des catégories. La figure 1.20 présente le système. La méthode de saisie a été testée pour des objets posés sur une table unie à laquelle est fixée le robot. Pour fonctionner, les positions relatives du laser et de la caméra doivent être parfaitement connues. Le temps de reconstruction et de traitement est significatif puisqu'il faut que le laser ait parcouru la totalité de la scène pour qu'elle puisse être reconstruite.

Une autre voie de recherche consiste à s'abstraire entièrement de l'utilisation de bases de données et à tenter de saisir des objets inconnus directement à partir des informations données en ligne par les capteurs. Un objet inconnu est un objet sur lequel on ne dispose pas d'informations *a priori*. La géométrie de l'objet peut être reconstruite précisément [Wang 05] ou partiellement en utilisant une nappe laser [Saxena 08] (voir figure 1.22) ou un système de caméras [Sahbahni 07] ou encore grossièrement approximée par des formes géométriques simples [Huebner 08b, Speth 08] pour en extraire des caractéristiques suffisantes à la saisie. L'utilisation de primitives simples est plus robuste aux erreurs de perception que la description précise de l'objet. Elle est également particulièrement appropriée à la saisie par une pince parallèle ou bien par un manipulateur par enroulement qui peut saisir une grande variété d'objets en ayant une connaissance minimale de leur caractéristiques géométriques [Dollar 06].



FIG. 1.19 – Le détecteur de points de saisie est entraîné sur des données simulées. Il est équipé d'une caméra qui lui permet de retrouver ces points de saisie sur les images courantes. Le système saisit ensuite les objets en boucle ouverte [Saxena 08].

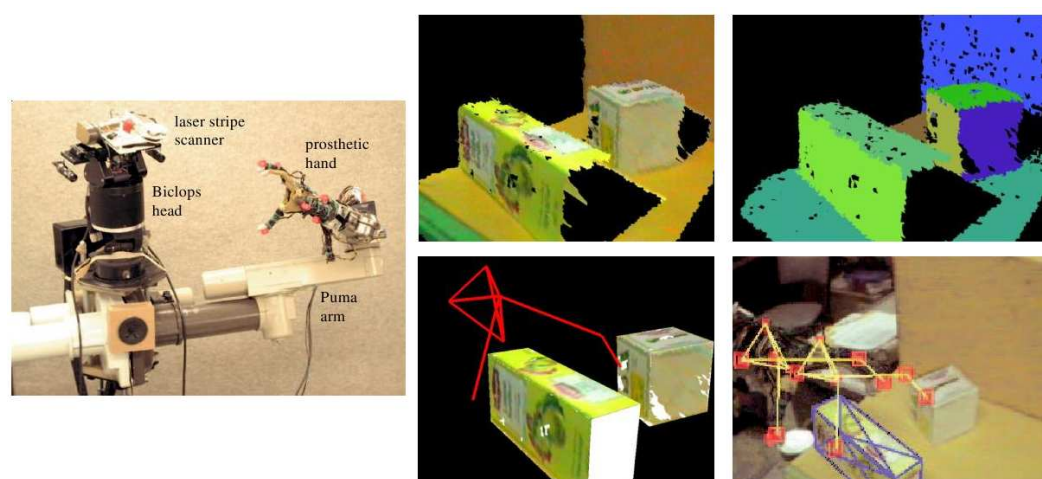


FIG. 1.20 – Un rayon laser balaye la scène et est détecté dans les images d'une caméra [Taylor 02].

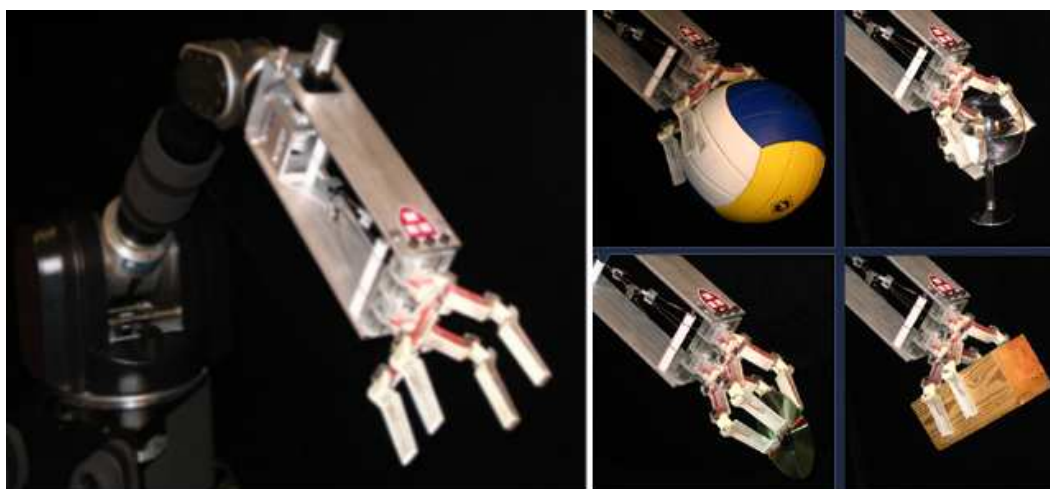


FIG. 1.21 – La main SDM permet de saisir une grande variété d'objets par enroulement [Dollar 06].



FIG. 1.22 – Saisie d'objets en connaissant partiellement leur forme [Saxena 08].

En suivant la voie ouverte par le scénario de MUSIC [Kazi 95], cité dans le paragraphe précédent, d'autres travaux tentent de résoudre le problème de la saisie d'objet en milieu humain en partageant judicieusement la commande entre l'utilisateur et le robot [Leroux 06, Remazeilles 08, Tsui 08]. La coopération entre l'humain et la machine permet alors à chacun de réaliser une tâche qu'il n'aurait pas pu réaliser seul [Kemp 07].

Le robot Aviso [Leroux 06, Remazeilles 08], qui est à la base de notre étude, est un bras Manus équipé d'un banc stéréo non calibré et d'une barrière optique en bout de pince. La stratégie de saisie

consiste à éviter les problèmes liés à la reconnaissance d'objet en ne considérant que des informations acquises en ligne par le système de vision et en proposant ainsi une méthode pour saisir des objets inconnus. Le scénario de saisie est le suivant : l'utilisateur déplace le bras en mode manuel jusqu'à ce que l'objet soit dans le champ de vision d'une des caméras, il sélectionne ensuite l'objet à saisir en cliquant deux fois sur la vue d'une des caméras. Ces deux *clics* définissent deux coins opposés de la boîte. Le bras approche ensuite de l'objet, pince parallèle au plan du sol, en centrant la boîte ainsi définie. Pour que ce scénario fonctionne, un certain nombre d'hypothèses doivent être vérifiées : les objets doivent être suffisamment petits pour pouvoir être saisis, ils sont positionnés de sorte à être frontaux parallèles aux caméras au moment de l'approche, ils doivent également être texturés, ce qui est une hypothèse très forte.

Le scénario proposé par le projet américain Halo [Tsui 08] est assez similaire au scénario Aviso, à cela près que les objets doivent être uniformes sur un fond relativement uniforme pour pouvoir être détectés. Halo est un bras MANUS monté sur un fauteuil roulant et équipé d'un système de vision. Une caméra est déportée sur une tête *pan tilt* et permet de voir l'ensemble de la scène ; une caméra est embarquée en bout de pince et donne une vue des détails de la scène. La saisie est déclenchée par un *clac* dans la vue globale et la pince approche l'objet en étant maintenue horizontale.

En allant encore plus loin dans ce sens, le projet ARPH propose une saisie d'objet en mode partagé, qui est en quelque sorte un mode *téléopéré intelligent*. Le robot est une plateforme mobile équipée d'un bras MANUS et de caméras. Il n'effectue que des tâches réflexes et toute l'intelligence est laissée à la personne qui le téléopère à distance en utilisant la vue des caméras embarquées sur le système. Le robot oriente les caméras de sorte à ce que l'utilisateur ait une perception plus intuitive lors de la commande du système. Une commande redondante des déplacements de la base et du bras permet d'augmenter la manipulabilité du système.

Dès lors que le système n'est pas entièrement téléopéré, la perception de l'environnement est nécessaire à la manipulation. La manipulation peut améliorer la perception, initiant un cercle vertueux. En poussant ou tirant sur des objets articulés et en détectant les mouvements dans les images acquises par un système de vision, les principales articulations de ces objets peuvent être identifiées et la meilleure manière de les saisir peut en être déduite [Katz 08]. La segmentation d'un objet est très complexe en milieu non structuré. En poussant un objet symétrique et en utilisant les images acquises avant et après la poussée, il est possible de segmenter cet objet même s'il est transparent et placé sur un fond complexe [Li 08]. La manipulation des objets peut permettre d'enrichir la modélisation de l'environnement.

Conclusion

La saisie d'objet en milieu humain pose des problèmes qui sont loin d'être résolus. Dès lors que l'environnement n'est pas contraint, le robot ne peut plus pratiquer de saisie autonome en aveugle, à la manière des robots industriels. Deux solutions sont alors envisageables : la saisie peut être entièrement manuelle, comme c'est le cas pour le robot TOU, ou le robot doit être équipé d'un ensemble de capteurs pour percevoir son environnement et adapter la saisie à la position et à la forme de l'objet. En d'autres termes, soit l'utilisateur perçoit l'environnement et contrôle les mouvements du robot, soit le robot est doté de sa propre capacité de perception afin de pouvoir agir en fonction de l'environnement sans intervention de l'utilisateur.

Il est toujours possible de simplifier le problème de la saisie en milieu humain en imposant des contraintes au système. Par exemple en limitant l'emplacement des objets à une zone connue, comme la table du projet CORA ou bien le plateau tactile du projet Friend. Une autre solution consiste à limiter l'ensemble des objets en les marquant ou bien en enregistrant leur apparence dans une base de données. Cependant, d'un point de vue pragmatique, limiter l'espace ou le nombre d'objets est très contraignant pour un usage quotidien.

L'idéal serait d'aller vers un outil robotique capable de saisir n'importe quel objet à partir des informations qui lui sont données en ligne par ses capteurs et de sa base de connaissance. Il ne

semble pas nécessaire de connaître précisément la géométrie d'un objet pour pouvoir le saisir. Mais quelles sont les primitives minimales qu'il est nécessaire d'extraire pour parvenir à la saisie d'un objet inconnu ? Est-il possible de saisir tous les types d'objets dans n'importe quelle scène ? Comment partager judicieusement la commande pour utiliser au mieux les capacités de l'utilisateur et le potentiel d'un système automatique ?

Dans cette étude, nous allons tenter de répondre à ces questions en proposant un outil d'assistance aux personnes lourdement handicapées pour la saisie d'objets inconnus en milieu humain. Notre démarche peut être vue comme une approche complémentaire aux approches utilisant des bases de données. Il serait dommage de ne pas utiliser de techniques éprouvées utilisant des bases de connaissance pour saisir des objets utilisés quotidiennement et de devoir les redécouvrir à chaque saisie. Par ailleurs, disposer d'une méthode permettant de saisir un nouvel objet ou un objet non reconnu est particulièrement intéressant dès que la personne sort de son domicile. Les méthodes de localisation et de caractérisation proposées dans cette thèse peuvent également servir à enrichir une base de données existante. Le stockage des informations et leur réutilisation ne seront pas traitées dans cette étude.

1.3 Proposition pour un outil de saisie contrôlé par la vision

Cette section décrit notre outil de saisie d'objets pour l'assistance aux personnes en perte d'autonomie. Notre approche sera présentée à plusieurs niveaux : tout d'abord le projet dont elle découle et sa philosophie, puis les capteurs permettant une perception de l'environnement, l'interface de commande, les hypothèses de travail et enfin les contributions de cette thèse.

1.3.1 Présentation générale

Cette thèse fait suite au projet AVISO qui a été mené par le CEA-LIST en partenariat avec l'AFM⁷, l'association APPROCHE⁸ et La Fondation Caisses d'Épargne pour la solidarité. Ce projet vise à simplifier la commande du bras MANUS en l'équipant d'un système de vision pour qu'il puisse être utilisé par des personnes lourdement handicapées. Il permet la saisie d'objets inconnus en limitant l'action de l'utilisateur à quelques *clics* sur une interface graphique. La figure 1.23 présente le robot AVISO durant une évaluation avec un patient dans les locaux du centre hospitalier de Garches, membre de l'association APPROCHE. L'originalité de ce projet réside dans l'interaction étroite entre les roboticiens, les équipes médicales et les patients et dans les efforts faits pour développer une interface de commande intuitive et adaptable au handicap de chaque patient tout en maintenant le coût du système relativement faible.

Ainsi, les développements effectués dans le cadre du projet AVISO sont en adéquation avec la plupart des critères mentionnés dans la première partie de ce chapitre. Le bras MANUS est un bras sécurisé qui peut être déployé à proximité des personnes. Les modifications apportées au système n'engendrent pas d'augmentation conséquente du coût du système ; le système de vision a été construit à partir de *webcams* achetées dans le commerce. Le travail sur l'interface de commande permet de connecter le périphérique le plus adapté au handicap de la personne pour utiliser l'interface graphique de commande. Le temps d'apprentissage nécessaire à l'utilisation du système est plus faible que le temps d'apprentissage nécessaire pour utiliser le bras MANUS standard et l'acceptation des patients est bonne, comme le montrent les résultats des évaluations avec les patients [Leroux 06].

Le scénario de saisie d'AVISO peut être décomposé en quatre temps (voir figure 1.24) :

1. Exploration et positionnement du bras

L'opérateur déplace le bras pour amener l'objet à saisir dans le champ de vision des caméras.

⁷ Association Française contre les Myopathies

⁸ Association pour la promotion des nouvelles technologies en faveur des personnes en perte d'autonomie



FIG. 1.23 – Aviso en cours d'évaluation au centre de Garches. Aviso est un bras MANUS équipé d'un banc de stéréovision et d'une interface homme machine intuitive

S'il ne voit pas l'objet, une phase préalable d'exploration est nécessaire. Le bras est commandé via une interface graphique [Leroux 04] ;

2. Désignation de l'objet

l'interface graphique permet d'afficher une vue de l'objet. La désignation de l'objet se fait par un encadrement au moyen de deux clics ;

3. Approche du bras

le bras approche l'objet et un asservissement visuel maintient la pince centrée sur la boîte, définie à l'étape précédente ;

4. Saisie

lorsque l'objet entre dans les mors de la pince, il coupe la barrière optique déclenchant la fermeture de la pince.

Pour saisir l'objet, l'utilisateur doit effectuer plusieurs sélections et validations sur l'interface graphique. Des tests du système dans des centres hospitaliers [Leroux 04] ont évalué le nombre de *clics* à 8 en moyenne. La phase la plus fatigante pour l'utilisateur est la phase de positionnement de bras qui demande plusieurs sélections.

Les hypothèses d'AVISO. Si AVISO permet de saisir des objets inconnus dans une scène inconnue, un certain nombre d'hypothèses sont émises sur les objets et leur pose :

- les objets sont texturés ;
- les objets sont verticaux ;
- les objets sont frontaux parallèles ;
- les objets sont bien séparés les uns des autres ;
- les objets peuvent être saisis par une approche de la pince horizontale et centrée sur la boîte désignée.
- les dimensions de l'objet lui permettent d'être détecté et saisi par la pince.⁹

Ces hypothèses restreignent l'*opérabilité* du système qui ne pourra pas saisir un objet qui ne respecte pas ces conditions. Une des applications demandées par les personnes en situation de handicap

⁹Cette hypothèse est communément admise dans les travaux de saisie d'objet.

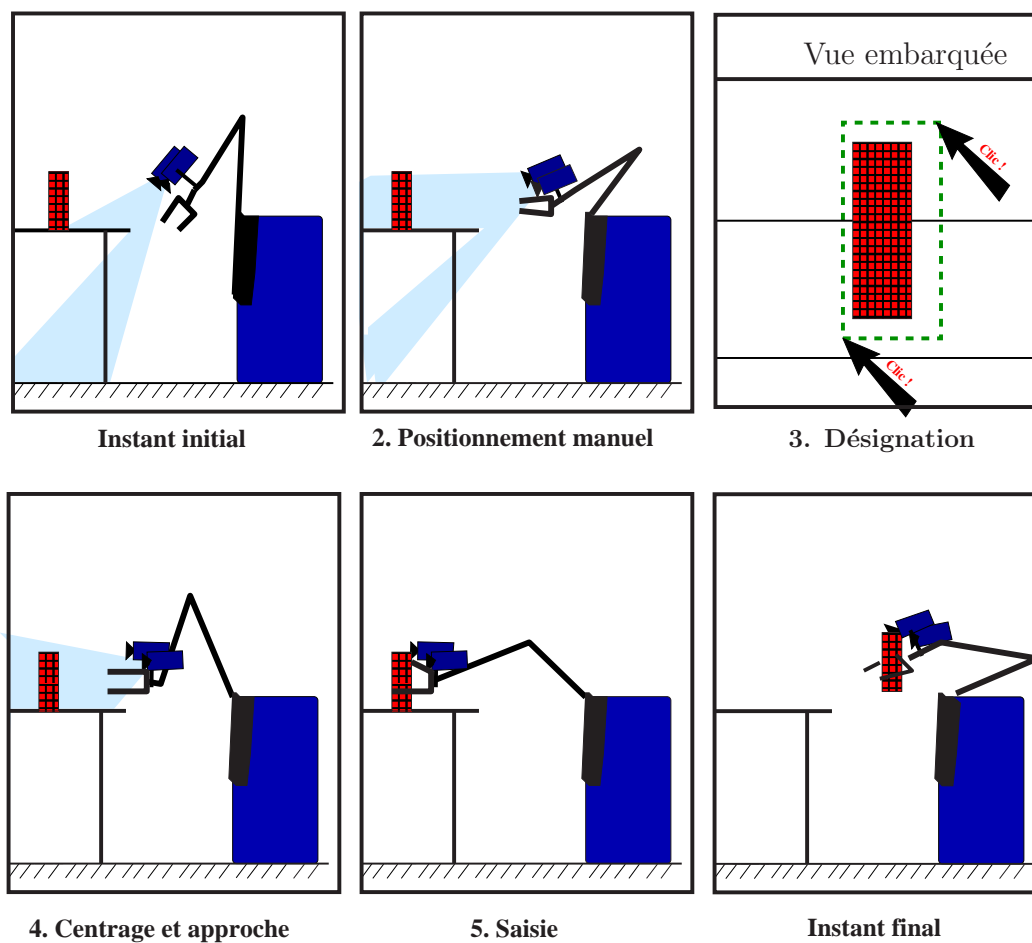


FIG. 1.24 – Le scénario de saisie Aviso

est de pouvoir saisir un objet dans une position quelconque posé sur une table ou une étagère ou encore ramasser un objet tombé au sol. C'est un des cas dans lequel l'outil de saisie AVISO est inefficace.

Objectifs de la thèse Cette thèse propose des méthodes pour étendre le scénario de saisie AVISO à des objets non texturés et en adaptant la saisie à la pose de l'objet. À terme, l'objectif est la saisie d'un objet en un seul *click* en conservant la philosophie du projet AVISO : les objets sont inconnus *a priori* et une intervention minimale de l'utilisateur est demandée. Dans le reste du document, nous nous référerons à cette approche comme l'approche *One click*.

Hypothèse 1 *Un point de l'objet est cliqué par l'utilisateur*

Les outils proposés sont conçus pour être utilisés dans un milieu non structuré où l'illumination est non contrôlée. L'objet à saisir peut se trouver sur une table couverte d'une nappe texturée, sur un support uniforme, au sol ou bien au milieu d'une scène complexe. Les cas d'un objet dans une boîte et d'un livre rangé dans une bibliothèque seront écartés.

La scène est supposée statique, c'est-à-dire la pose des objets ne varie pas dans le temps. Une première hypothèse est donc imposée¹⁰ :

Hypothèse 2 *La scène est statique*

Le robot devra pouvoir saisir un large éventail d'objets sans utiliser de marqueurs, de bases de données d'apparences ni de bases de données de modèles [Saxena 08]. Le système ne dispose donc pas d'informations *a priori* sur les objets, ils sont alors qualifiés *d'objets génériques*. Un *objet générique* est un objet sur lequel aucune information n'est disponible *a priori* sur sa texture, sa forme, sa taille, son apparence. Contrairement au projet AVISO, aucune contrainte ne sera imposée sur la position des objets qui sera un des paramètres à déterminer pour parvenir à la saisie. L'objet à saisir est dans une position quelconque.

Hypothèse 3 *Les objets sont génériques*

L'étude sera limitée à des objets rigides et d'un seul tenant. La saisie d'objets articulés ou déformables ne sera pas traitée dans cette étude.

Hypothèse 4 *Les objets sont rigides*

L'objectif de cette thèse est donc de conserver la philosophie du projet AVISO (des objets génériques, une intervention minimale de l'utilisateur, une interface intuitive) tout en ouvrant son utilisation à un plus large éventail d'objets et en poussant la simplification de son interface de commande à son minimum en proposant une commande *One click*. Le scénario de saisie et le système seront modifiés pour parvenir à cet objectif.

1.3.2 La perception de l'environnement

La saisie d'un objet dont la forme et la pose sont inconnues *a priori* nécessite des capacités de perception de l'environnement pour détecter l'objet, le localiser, l'approcher et le saisir. Les systèmes de vision sont séduisants de part la richesse des informations qu'ils permettent d'obtenir, le faible coût des caméras et leur faible encombrement. Ils sont massivement utilisés par les systèmes de saisie d'objets existants, en particulier lorsque l'environnement est connu ou lorsque l'objet est connu.

Les outils de saisie d'objets inconnus en milieu inconnu recourent souvent à la fusion de données de différents capteurs et en particulier en utilisant à la fois un capteur actif de type laser et un capteur passif de type vidéo. Ces deux types de capteurs sont complémentaires : les caméras fournissent une information 2D dense tandis que les capteurs actifs donnent une information sur la profondeur.

¹⁰Les hypothèses émises sur le système et sur la scène seront égrainées au fil de ce manuscrit. L'ensemble des hypothèses est rappelé dans l'annexe C

Nous avons choisi d'utiliser uniquement des caméras, principalement pour des raisons de définition, d'encombrement mais aussi pour des raisons de coût. Le plus petit capteur laser pèse un peu plus 250g. Or le bras MANUS que nous utilisons ne peut porter que 500g en bout de bras. Ainsi, équiper le bras d'un tel capteur utiliserait pratiquement la totalité la charge utile et limiterait la saisie à de tout petits objets de poids inférieurs à 100g. Il est alors possible d'envisager de le fixer sur un support extérieur. Cependant, la résolution du système est de plusieurs centimètres à 1m. Ainsi, des petits objets ne pourraient pas être saisis. Enfin, ce système laser vaut un peu plus de 1000 euros alors que la caméra que nous avons monté en bout de bras est une webcam standard qui vaut moins de 50 euros et pèse moins de 50g.

Les capteurs lasers sont de plus en plus couramment utilisés en robotique ce qui accroît les recherches pour leur miniaturisation et la densification de leurs données. Ces arguments de coût, de poids et de définition ne seront donc peut-être plus valables dans peu de temps. Par exemple, le capteur actif développé par l'EPFL pourrait être particulièrement intéressant dans le cadre de notre application de saisie d'objet. En particulier, couplé à une caméra, il peut permettre de segmenter efficacement l'objet sur le fond.

L'objectif de cette thèse est de saisir des *objets génériques* positionnés de manière quelconque dans une scène inconnue. Nous écarterons les situations dégénérées dans lesquelles un système de vision ne peut segmenter un objet du fond, typiquement, le cas d'un objet uni du même ton que l'arrière plan uni sur lequel il se trouve ne sera pas traité.

Hypothèse 5 *Les objets sont détectables en utilisant la vision*

L'utilisation de la vision pose la question du positionnement des caméras pour récolter les informations nécessaires à la saisie. Plusieurs montages sont envisageables : les caméras peuvent être montées sur le système lui-même ou directement fixées à l'environnement (sur un trépied ou sur le plafond ou les murs d'une pièce, par exemple). Lorsqu'elles sont fixées sur le système, elles peuvent être fixées à la carlingue du robot, montées sur une tête *pan-tilt*, ou bien encore fixée sur l'effecteur d'un manipulateur. Selon leur degré de mobilité et leur distance à la scène chacun de ces montages présente des caractéristiques particulières qui permettent d'accéder à des informations différentes. Une caméra fixée au plafond d'une pièce ou d'un hall donne une vue large qui peut contenir une grande partie de la pièce, ou du hall, le système robotique, et la scène où se trouve l'objet. À l'inverse, une caméra fixée sur le système lui-même donne généralement une vue plus réduite de la scène. Cette vue suit les déplacements du robot. Enfin, une caméra embarquée sur l'effecteur est mobile relativement au robot et à la pièce. Elle peut être déplacée à l'intérieur même de la scène et donne une vue plus précise des éléments qui la composent.

Il est également possible de combiner plusieurs montages afin de cumuler les avantages de chacun d'entre eux. Par exemple, en utilisant ces systèmes séquentiellement, il peut être intéressant de disposer d'une vue globale de la pièce pour déplacer le robot jusqu'à une table ou une étagère où se trouve l'objet à saisir puis d'une vue centrée sur la scène pour localiser l'objet et enfin d'une vue embarquée sur la pince pour acquérir suffisamment de détails pour exécuter une saisie fine. Il est également intéressant d'utiliser ces systèmes simultanément pour commander séparément différents degrés de liberté d'un robot. Par exemple, en utilisant une caméra déportée pour commander les translations de l'effecteur d'un manipulateur et une caméra embarquée sur la pince pour commander les rotations [Flandin 00].

Dans le cas d'un système d'assistance à la saisie, une attention particulière doit être portée à la limitation des installations. Il est préférable de ne pas contraindre l'environnement et de limiter les lieux de fixation des caméras au système lui-même. Nous avons choisi d'utiliser un système composé de deux caméras, l'une portée par la pince et l'autre fixée à la plateforme pour guider la saisie de l'objet (voir figure 1.25).

Quelle que soit la configuration considérée, il est courant de supposer que l'objet à saisir est dans le champ de vision de toutes les caméras composant le système avant le lancement de la tâche de vision. Soit la position de chacune des caméras est fixée et l'objet est compris dans l'intersection des champs de vision, soit le placement de l'objet est libre et une phase de positionnement des caméras, préalable à la saisie, est nécessaire. La plupart des travaux ne mentionnent pas cette étape de positionnement et très peu d'études ont été consacrées à son automatiser. Elle est généralement laissée au soin de l'utilisateur qui doit déplacer les caméras en mode manuel avant de lancer la saisie, comme le proposent les projets AVISO et ANSO [Leroux 07]. Cette phase de positionnement peut être lourde pour une personne en situation de handicap, voire impossible si cette personne ne dispose pas d'une dextérité suffisante à la téléopération du système. Nous proposerons dans la première partie de cette étude une solution au positionnement du capteur embarqué à partir d'une seule désignation dans la vue déportée.

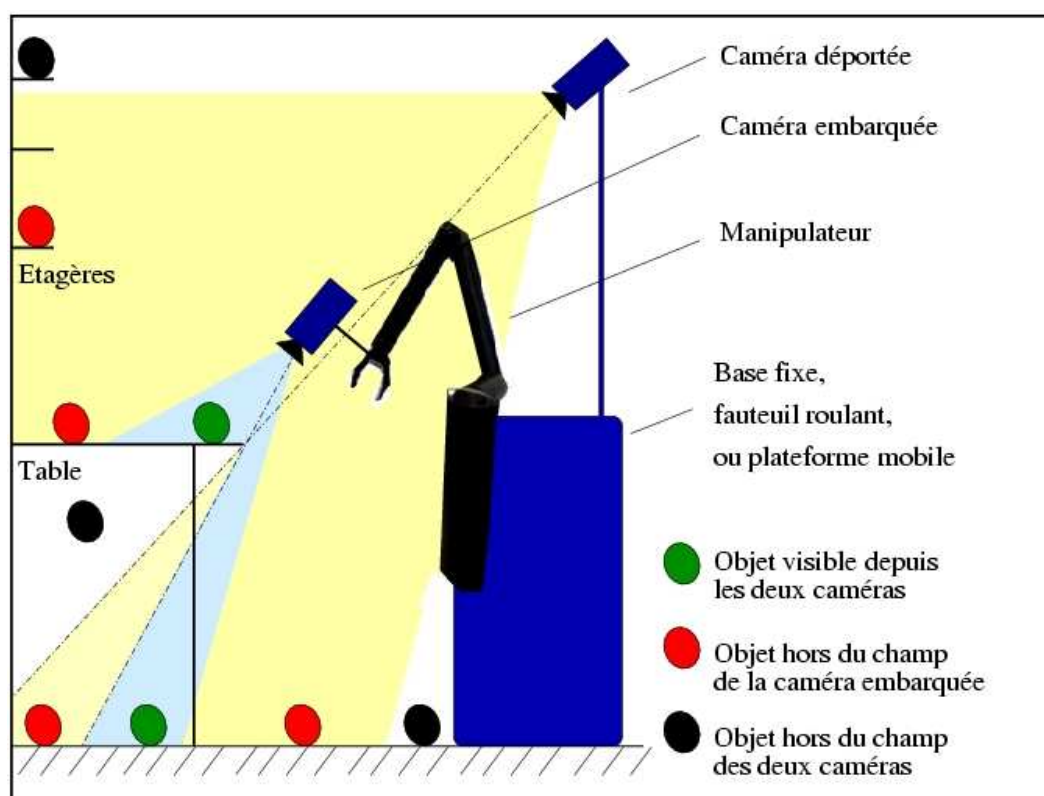


FIG. 1.25 – Coopération caméra déportée - caméra embarquée : les disques verts, rouges et noirs représentent les différentes situations rencontrées en fonction de la position de l'objet à saisir, c'est-à-dire s'il est vu respectivement par les deux caméras, par la caméra déportée uniquement ou s'il n'est visible par aucune des caméras.

La figure 1.25 illustre les différentes situations rencontrées selon la position qu'occupe l'objet dans l'espace. L'objet est supposé se trouver dans la zone de travail du bras. Le positionnement préalable du bras, garantissant cette hypothèse, peut être effectué en mode manuel que cela soit par un déplacement du fauteuil ou de la plateforme commandée par l'utilisateur ou en mode automatique, par exemple, en utilisant la redondance entre le bras et le mobile [Colle 08]. Dans ce travail, le bras est en position finale de saisie.

Hypothèse 6 *Les objets sont dans la zone de travail du bras*

La caméra déportée dispose d'un champ de vue suffisamment large pour couvrir toute la zone de travail du bras. Si l'hypothèse 6 est vérifiée, alors l'objet se trouve dans le champ de vision de la caméra déportée. Les positionnements non traités sont symbolisés par un disque noir dans la figure 1.25, ils correspondent à des objets hors du champ de vision de la caméra déportée.

Hypothèse 7 *Les objets sont dans la vue déportée*

1.3.3 La désignation de l'objet

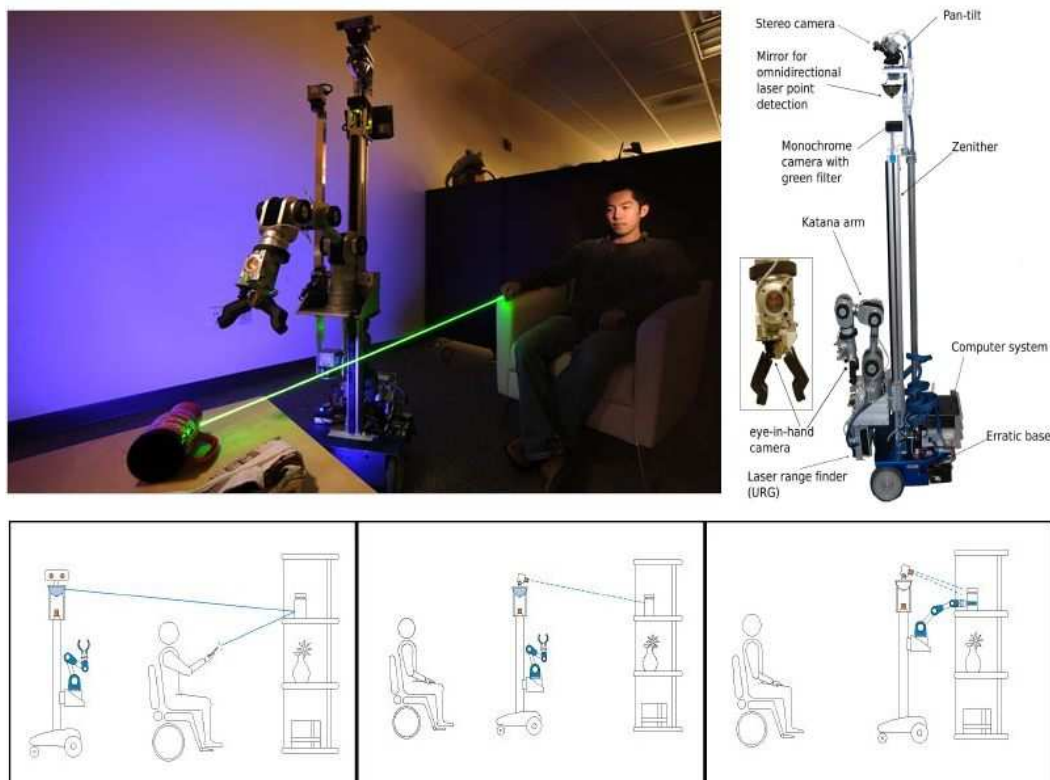


FIG. 1.26 – Le projet EI-E : à gauche, l'objet à saisir est pointé par un laser vert. Le robot détecte le point vert résultant dans l'image d'une caméra omnidirectionnelle. Au milieu, le robot oriente sa tête stéréovision en direction du point détecté et estime la position 3D du point. À droite, le robot va saisir l'objet [Nguyen 08].

Pointer un objet est une manière intuitive de le désigner, il semble donc naturel de proposer une interface homme-robot basée sur un pointage pour désigner une action à effectuer. Trois catégories de pointages peuvent être distinguées [Edsinger 06, Kemp 07]. Tout d'abord, le *clic* sur une interface graphique qui affiche, par exemple, une vue de la scène [Leroux 06, Tsui 08]. Viennent ensuite les désignations dites "*naturelles*" : l'utilisateur montre une direction en pointant l'objet à la main [Becker 98, Becker 99] ou bien guide le robot à la voix. Enfin l'utilisation d'outils avancés comme un pointeur laser [Kazi 95, Kemp 07, Nguyen 08], ou bien encore les outils *Xwand* et *world cursor* (voir figure (1.27) et (1.26)) qui consistent en une baguette équipée de capteurs inertiels et d'un laser dans un environnement complètement équipé. De l'utilisation de ces interfaces est déduite une direction qui porte l'objet ou bien directement une localisation 3D.

Si la désignation *naturelle* ou via des outils de pointage est intuitive et permet à une personne valide de désigner simplement une partie de la scène, elle n'est pas triviale pour une personne dont

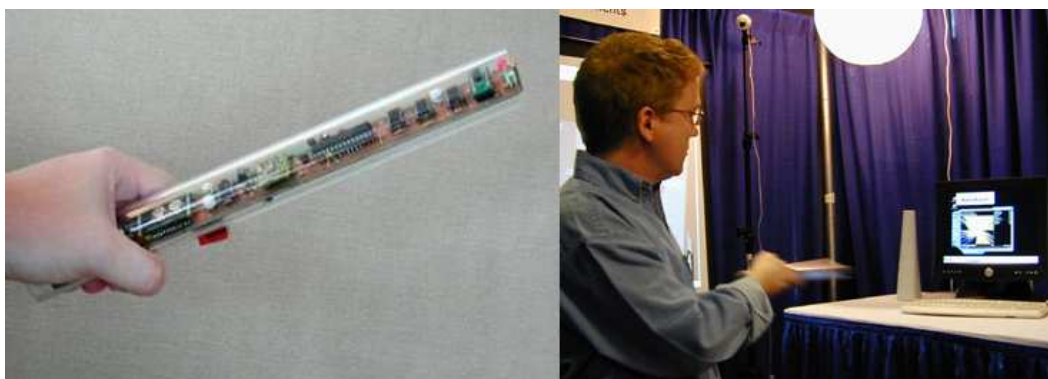


FIG. 1.27 – La baguette *xwand* contient une centrale inertielle et un laser. Elle permet de pointer un périphérique d'une maison intelligente et d'agir sur son état : allumer la télévision, allumer ou éteindre une lampe, etc ...

la mobilité est réduite. À l'opposé, il existe une large gamme de périphériques permettant de générer un *clac* sur un écran pour tout type de handicap [Leroux 04]. Une commande par désignation sur une interface graphique est donc plus adaptée à un outil destiné aux personnes handicapées.

Cependant, la commande par *clacs* sur une interface graphique affichant une vue de la scène, oblige l'utilisateur à changer de perspective entre ce qu'il voit et ce qui est vu depuis les caméras. Cela rend cette interface un peu moins intuitive que les désignations naturelles et pose le problème du positionnement des caméras pour que l'utilisateur puisse facilement interpréter et utiliser les images qu'elles acquièrent. Ainsi, si l'utilisateur a suffisamment de mobilité pour désigner un objet par une *désignation naturelle* le temps d'apprentissage peut être réduit en lui proposant ce type d'interface. Dans le cas contraire, une interface par *clacs* offre une alternative aux handicaps les plus sévères.¹¹

Dans ce travail, nous avons étudié une interface basée sur une désignation par un seul *clac* sur une vue globale de la scène. L'information fournie est une direction de l'espace reliant le centre optique de la caméra au *clac* (voir chapitres suivants). Cette information est identique à celle qui pourrait être extraite d'une désignation naturelle par pointage à la main ou à l'aide d'un outil de pointage. Les méthodes développées dans cette thèse sont directement applicables à ce type de système. Dans le cas d'une désignation à la voix, la première étape consiste à *reconnaître* l'objet désigné, ce qui nécessite l'utilisation d'une base de données contenant des informations sur les objets qui peuvent être saisis par le système. L'un des objectifs de cette thèse est de s'abstraire de l'utilisation de base de données ; les méthodes que nous avons développées ne sont donc pas destinées à être utilisées avec ce type de désignation.

1.3.4 Le scénario de saisie

L'objectif de cette thèse est de tendre vers une application *One Click*, c'est-à-dire qui ne nécessite qu'une seule validation de l'utilisateur. Pour parvenir à une saisie en un seul *clac* notre scénario de saisie est également décomposé en quatre étapes qui sont illustrées sur la figure (1.28) :

1. Désignation

À l'instant initial, aucune information n'est disponible sur l'objet et la scène. L'opérateur désigne l'objet sur une vue d'ensemble de la scène par un *clac* et limite ainsi l'ensemble des positions possibles de l'objet à une ligne de l'espace ;

2. Localisation

En utilisant l'information donnée par le *clac*, la position de l'objet sur la ligne de vue associée est déterminée. Cette étape donne une estimation grossière de la localisation de l'objet dans l'espace, cependant cette information n'est pas suffisante pour saisir l'objet ;

¹¹Certaines personnes ne peuvent effectuer aucune de ces actions, dans cas on préférera utiliser un système de balayage séquentiel (chenillard) avec un seul contacteur.

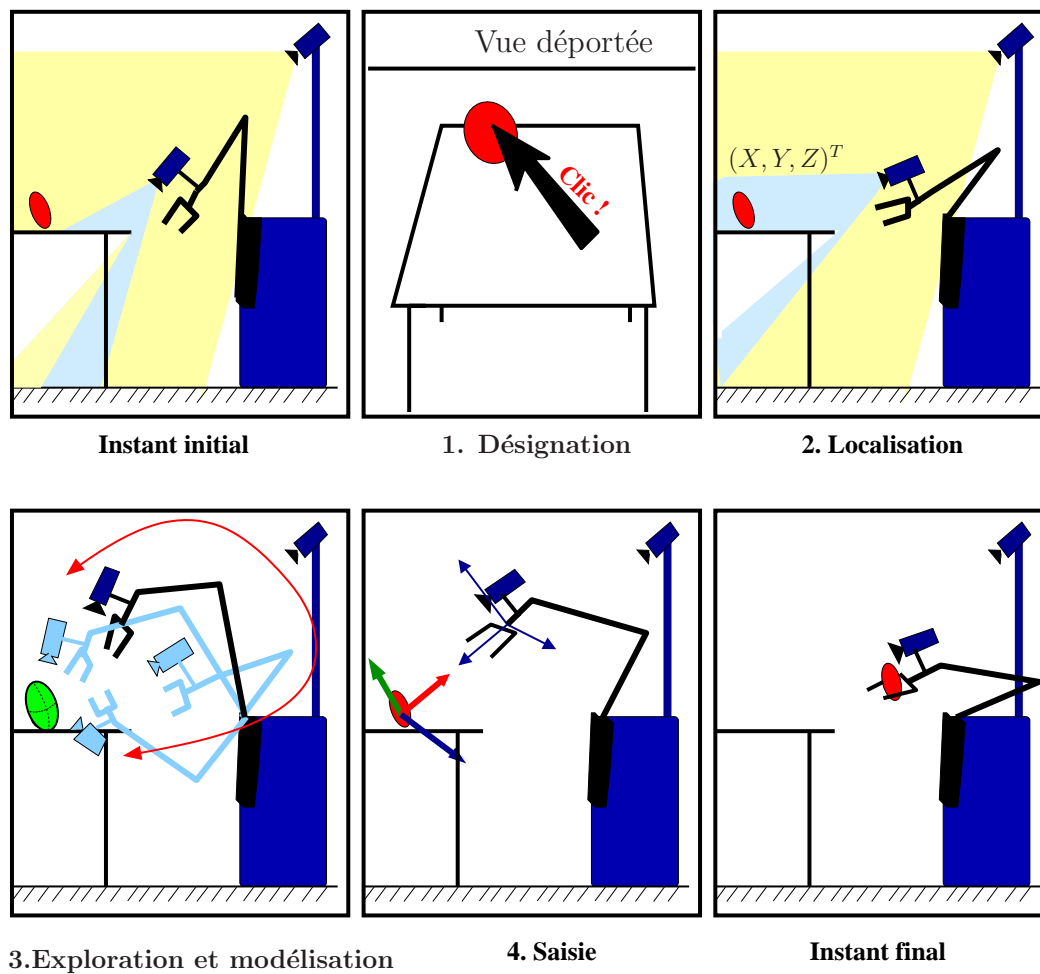


FIG. 1.28 – Le scénario de saisie One Click

3. Exploration et modélisation :

La caméra embarquée permet d'explorer l'objet depuis différents points de vue. Ces observations permettent d'estimer la forme et la pose de l'objet. Ces informations sont suffisantes pour saisir un objet sous certaines hypothèses qui seront détaillées dans ce manuscrit ;

4. Approche et Saisie

Finalement, les estimations de la pose et de la forme de l'objet sont utilisées pour approcher la pince de manière adéquate. Lorsque l'objet entre dans les mors de la pince, il coupe une barrière optique montée à l'intérieur des mors et déclenche la fermeture de la pince. Des capteurs d'efforts montés à l'intérieur de la pince permettent d'ajuster la force de serrage pour soulever l'objet sans l'abîmer.

1.3.5 Les contributions de cette thèse

Cette thèse propose des outils pour permettre la saisie d'un objet inconnu à partir d'un *clic* de l'utilisateur sur une vue globale de la scène dispensée par une caméra déportée. Aucune connaissance *a priori* n'est disponible sur les objets à saisir et sur la scène, c'est-à-dire il n'y a pas eu de phase préalable de construction de base de données ni de marquage. Les objets à saisir sont vus pour la première fois. Les principales contributions de cette étude concernent les étapes de localisation, de modélisation et d'exploration de la scène pour permettre la saisie d'un nouvel objet.

L'étape la plus coûteuse du scénario AVISO, en terme d'interactions homme - machine, est, sans conteste, l'étape de positionnement de la caméra embarquée qui permet de visualiser et de localiser l'objet. La première contribution de cette thèse est une méthode de **localisation d'un objet inconnu** en utilisant uniquement l'information donnée par le *clic* et les informations données en ligne par deux caméras. Nous verrons comment utiliser simultanément les informations données par la caméra embarquée et la caméra déportée pour estimer la position de l'objet et finalement positionner la caméra mobile de façon à ce que l'objet soit dans son champ de vision.

La seconde contribution de cette thèse est une **modélisation minimale d'objet inconnu** permettant d'adapter l'approche de la pince à sa forme et à ses dimensions. Connaissant la position de l'objet, il est possible de déplacer la caméra embarquée en bout de pince de sorte à acquérir plusieurs vues de l'objet jusqu'à ce que l'information obtenue soit suffisante à sa saisie. Nous définirons quel est le type d'information nécessaire et enfin ce que signifie une représentation suffisante.

La troisième contribution est une **méthode de vision active permettant de sélectionner par les meilleures vues** pour affiner la représentation de l'objet. Cette méthode repose sur la quantité d'information, au sens de l'information de Shannon, contenue dans chaque vue. Nous montrerons que la quantité d'information contenue dans la vue suivante peut être directement prédite à partir des résultats courants de la reconstruction.

Ces méthodes peuvent être vues à la fois comme une alternative ou un complément des méthodes reposant sur des bases de connaissance. En effet, si ces méthodes ont déjà fait leurs preuves dans des applications de saisie d'objet en milieu humain, elles échouent lorsqu'un objet n'est pas reconnu à cause d'un défaut de perception ou s'il est vu pour la première fois. Dans le cas d'un tel échec, les méthodes que nous proposons peuvent être utilisées pour saisir l'objet non reconnu et mettre à jour la base de données en y ajoutant ses caractéristiques. À l'inverse, plutôt que de redécouvrir à chaque saisie un objet que l'on observe fréquemment, il pourrait être intéressant de stocker les informations obtenues à la première saisie, afin de les utiliser lors des observations ultérieures, pour gagner du temps sur les phases d'exploration. Cette partie ne sera pas traitée dans cette thèse et sera laissée en perspective.

Les quatre grandes étapes du scénario de saisie structurent ce manuscrit : i) l'utilisateur désigne l'objet par un *clic*, ii) l'objet est localisé puis iii) la caméra embarquée observe l'objet pour le mo-

déliser et enfin, iv) l'objet est saisi en adaptant l'approche de la pince à sa pose et à sa forme. Les principales contributions de cette thèse font l'objet des chapitres ultérieurs. La première partie traite de la **localisation d'un objet inconnu** à l'aide d'un système de deux caméras et à partir d'un *clic*. La partie suivante traite de la **modélisation de l'objet** et de **l'exploration de la scène** en utilisant la caméra embarquée afin de définir la meilleure stratégie de préhension.

L'objectif est de montrer qu'un manipulateur équipé de deux caméras peut saisir un objet inconnu désigné par un *clic* sur une vue de la scène.

Première partie

One Click Focus : estimation de la position d'un objet inconnu à partir d'un *clic*

À l'origine, l'outil d'aide à la saisie disponible préalablement à cette étude, était équipé d'un banc de stéréovision monté sur la pince d'un bras manipulateur [Leroux 06, Remazeilles 08]. La première étape du scénario de saisie consistait alors à positionner la pince, en mode manuel, afin que l'objet d'intérêt entre dans le champ de vision des caméras. L'objet était ensuite sélectionné par un encadrement de deux clics (voir le scénario de saisie sur la figure 1.24). Puis une phase de saisie automatique était lancée.

Dans le cadre de cette étude, nous avons choisi de modifier l'emplacement des caméras en proposant d'utiliser une seule caméra en bout de pince et une caméra déportée, permettant de sélectionner l'objet par un *clic* dans une vue globale de la scène. L'objectif de cette première partie est de proposer une méthode pour estimer la position de l'objet inconnu à partir de l'unique information donnée par un *clic* de l'utilisateur dans une image et en respectant les hypothèses établies à la fin du premier chapitre et rappelées en annexe C : la scène est statique, les objets sont génériques, rigides et détectables en utilisant un système de vision, les objets se situent dans la zone de travail du bras qui est contenue dans la vue de la caméra déportée.

L'outil de saisie que nous proposons repose sur une commande visuelle. La première section de ce chapitre rappelle donc les fondements théoriques de la vision par ordinateur et présente le problème de l'estimation de la position d'un objet dans le cadre de notre application. Les chapitres à venir présentent deux contributions de cette thèse :

- **Une commande basée sur la géométrie épipolaire** qui permet de parcourir la ligne de vue sur laquelle se trouve l'objet ;
- **L'estimation dynamique de la position de l'objet** à partir des images acquises au cours du parcours de la ligne de vue.

Une fois l'objet grossièrement localisé, la caméra embarquée sera déplacée de sorte à ce que l'objet soit dans son champ de vision. À la fin de cette étape, deux vues de l'objet seront donc disponibles et utilisables pour caractériser sa forme tridimensionnelle.

Représentation géométrique

L'outil de saisie *One Click* que nous proposons est un robot manipulateur qui perçoit son environnement via deux caméras : l'une est montée sur le support du bras et donne une vue globale de la scène, et l'autre, montée sur la pince, offre une vue des détails.

Pour estimer la position d'un objet à partir d'un système de vision, il est nécessaire d'établir la relation entre le monde tridimensionnel (3D) et sa représentation bidimensionnelle (2D) dans une image acquise par une caméra.

Ce chapitre rappelle les fondements théoriques de la géométrie perspective pour une, puis deux caméras. Elle présente également les principales notations qui sont associées à notre système. Ensuite la stratégie adoptée pour permettre la localisation d'un objet à partir d'un seul *clik* est présentée. Les méthodes permettant de la mettre en œuvre seront développées dans les chapitres suivantes.

2.1 Rappels mathématiques et géométriques

En vision par ordinateur, la géométrie projective est préférée à la géométrie euclidienne car elle permet de rendre linéaire une grande partie des transformations de l'espace euclidien, notamment la projection centrale qui est le modèle de projection le plus souvent utilisé.

Dans ce paragraphe, nous montrerons comment exprimer la position et l'orientation, autrement dit la *pose*, d'un objet ou d'une caméra dans l'espace euclidien, noté \mathbb{E}^3 et dans l'espace projectif, noté \mathbb{P}^3 .

2.1.1 De l'espace euclidien à l'espace projectif.

En général, les points de l'espace 3D ne sont pas directement exprimés dans le repère de la caméra mais dans un repère différent, généralement le repère de référence \mathcal{R}_b ou dans le repère attaché à l'objet \mathcal{R}_o . La transformation entre le repère de la caméra et un autre repère d'expression est appelée la *pose* de la caméra. Dans le reste de ce document, elle sera exprimée par une matrice homogène. La pose d'un solide, un objet ou une caméra, est la description de sa position et de son orientation dans un repère de référence. Elle permet par exemple de transformer les coordonnées d'un point d'un repère donné à un repère de référence. Son expression sera présentée dans l'espace euclidien, puis dans l'espace projectif.

2.1.1.1 Expression euclidienne de la pose.

Soit \mathcal{X} un point 3D. En géométrie euclidienne, sa position dans le repère orthonormé direct \mathcal{R}_b est notée sous la forme d'un vecteur ${}^b\bar{X}$ à trois composantes selon les axes cartésiens \mathbf{i} , \mathbf{j} et \mathbf{k} :

$${}^b\bar{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.1)$$

Ces coordonnées sont appelées les *coordonnées cartésiennes* d'un point.

Soit \mathcal{S} un solide. En plus de sa position, l'expression de son orientation peut être établie. Elle peut être exprimée par une matrice de rotation ${}^b\mathbf{R}_s$, c'est-à-dire la rotation du solide dans le repère de base. C'est une matrice orthogonale directe, ce qui signifie que ses colonnes forment une base orthonormée directe de l'espace vectoriel euclidien \mathbb{E}^3 , ou encore :

$${}^b\mathbf{R}_s^T {}^b\mathbf{R}_s = \mathbf{I}$$

$${}^b\mathbf{R}_s {}^b\mathbf{R}_s^T = \mathbf{I}$$

$$\det({}^b\mathbf{R}_s) = 1$$

Il existe plusieurs manières de construire la matrice de rotation. Par exemple, l'orientation de \mathcal{S} par rapport à \mathcal{R}_b peut être exprimée en écrivant les vecteurs unitaires qui portent ses axes principaux dans le repère \mathcal{R}_b . La *matrice de rotation* ${}^s\mathbf{R}_b$ est obtenue en concaténant ces trois vecteurs qui sont les trois projections d'une base orthonormée dans une autre.

Dans l'espace euclidien, un changement du repère d'expression d'un point, par exemple du repère \mathcal{R}_s au repère \mathcal{R}_b s'exprime de la manière suivante :

$${}^b\bar{X} = {}^b\mathbf{t}_s + {}^b\mathbf{R}_s {}^s\bar{X} \quad (2.2)$$

où ${}^b\mathbf{t}_s$ est le vecteur de translation et ${}^b\mathbf{R}_s$ est la matrice de rotation qui permettent d'exprimer ${}^b\bar{X}$ dans \mathcal{R}_b à partir de ${}^s\bar{X}$ dans \mathcal{R}_s .

Dans l'espace euclidien de dimension 3, les matrices de rotation suivantes correspondent à des rotations autour des axes $\mathbf{i}, \mathbf{j}, \mathbf{k}$ (respectivement) :

$${}^s\mathbf{R}_b(\mathbf{i}, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

$${}^s\mathbf{R}_b(\mathbf{j}, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$${}^s\mathbf{R}_b(\mathbf{k}, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

et la matrice de rotation ${}^b\mathbf{R}_s$ peut s'exprimer de différentes manières par composition de ces trois matrices, selon la convention choisie (angles d'euler, roulis tangage lacet, etc.). Par exemple, ${}^b\mathbf{R}_s = {}^s\mathbf{R}_b(\mathbf{i}, \theta_i) {}^s\mathbf{R}_b(\mathbf{j}, \theta_j) {}^s\mathbf{R}_b(\mathbf{k}, \theta_k)$.

2.1.1.2 Expression projective de la pose.

Un point \mathcal{X} de l'espace 3D défini par ses coordonnées cartésiennes $\bar{\mathbf{X}} = (X, Y, Z) \in \mathbb{E}^3$ peut être décrit dans l'espace projectif par ses *coordonnées homogènes* $\mathbf{X} = (\bar{\mathbf{X}}, 1) = (\lambda\bar{\mathbf{X}}, \lambda) \in \mathbb{P}^3$ où $\lambda \in \mathbb{R}$. L'espace projectif \mathbb{P}^3 peut être vu comme une extension de l'espace euclidien \mathbb{E}^3 . Un point sera décrit de manière équivalente par un vecteur $\bar{\mathbf{X}}$ à 3 dimensions dans \mathbb{E}^3 et par un vecteur \mathbf{X} à 4 dimensions dans \mathbb{P}^3 .

Le passage de l'espace projectif à l'espace euclidien s'effectue en divisant les coordonnées \mathbf{X} par λ . Si $\lambda = 0$ alors le point représente un point à l'infini qui est défini spécifiquement dans \mathbb{P}^3 et n'appartient pas à \mathbb{E}^3 . Ce point représente le point infini qui modélise l'intersection de droites parallèles à l'infini. Ce point représente la direction de ces droites parallèles dans l'espace euclidien.

La géométrie projective permet d'obtenir une expression homogène de l'équation 2.2 qui devient :

$${}^b\mathbf{X} = {}^b\mathbf{M}_s {}^s\mathbf{X} \quad \text{avec } {}^b\mathbf{M}_s = \begin{bmatrix} {}^b\mathbf{R}_s & {}^b\mathbf{t}_s \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (2.3)$$

Le groupe des transformations homogènes ${}^b\mathbf{M}_s$ est un sous ensemble des transformations projectives de \mathbb{P}^3 dans \mathbb{P}^3 . De manière générale, les transformations projectives de \mathbb{P}^n dans \mathbb{P}^n sont appelées *homographies* ou *collinéations*. Une matrice d'homographie étant inversible, la transformation inverse ou *duale* de ${}^s\mathbf{M}_b$ est ${}^b\mathbf{M}_s$ telle que :

$${}^s\mathbf{M}_b = {}^b\mathbf{M}_s^{-1} = \begin{bmatrix} {}^b\mathbf{R}_s^\top & -{}^b\mathbf{R}_s^\top {}^b\mathbf{t}_s \\ 0_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} {}^s\mathbf{R}_b & {}^s\mathbf{t}_b \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (2.4)$$

Ce formalisme est transitif, le transfert d'un repère à un autre peut donc s'exprimer en passant par tout un jeu de repères intermédiaires :

$${}^a\mathbf{X} = {}^a\mathbf{M}_b {}^b\mathbf{X} = {}^a\mathbf{M}_d {}^d\mathbf{M}_c {}^c\mathbf{M}_b {}^b\mathbf{X} \quad (2.5)$$

Notons que par souci de notation, nous ne ferons plus de différence entre les coordonnées cartésiennes ${}^a\bar{\mathbf{X}}$ et les coordonnées homogènes associées ${}^b\mathbf{X}$ d'un point 3D. Dans le reste de ce document, nous utiliserons principalement la notation homogène. Les quelques fois où la notation euclidienne sera utilisée seront précisées.

2.1.2 Le modèle sténopé

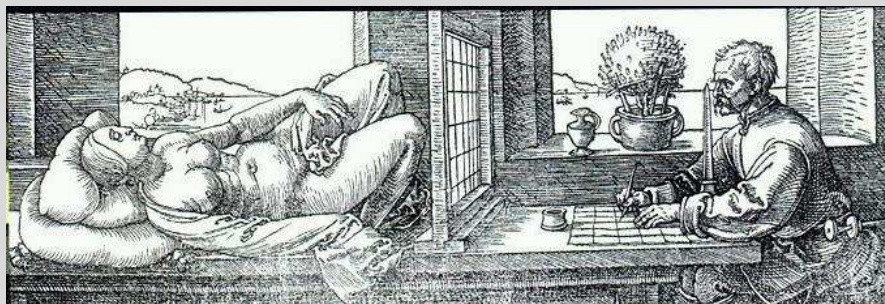
Les capteurs visuels tels l'œil, les caméras ou encore les appareils photographiques, sont dit *passifs*, par opposition aux capteurs dits *actifs* tels le scanner ou le laser. Ils sont essentiellement composés d'une optique, constituée de lentilles et de diaphragmes, d'un boîtier (ou boîte noire) et d'une surface sensible à la lumière. Dans le cas des caméras numériques, la surface sensible est une matrice plus ou moins dense de capteurs élémentaires, sensibles aux photons. Ce paragraphe décrit la projection perspective qui permet la formation des images.

2.1.2.1 Projection perspective ou modèle sténopé

Classiquement, en vision par ordinateur, la modélisation *sténopé*, ou trou d'épingle *pin-hole* (voir encadré 2.2) est substituée à la modélisation des lentilles minces [Baker 98]. Géométriquement, le modèle sténopé repose sur une projection dite *centrale* ou *perspective*. Pour tout point du monde tridimensionnel, un rayon passant par un point fixe dans l'espace est tracée. Ce point fixe est appelé *centre de projection*.

Les droites intersectent un plan particulier appelé *plan image*, formant ainsi l'image 2D \mathbf{x} du point 3D \mathcal{X} . L'expression de \mathbf{x} en unités métriques sera notée indifféremment, par la suite, \mathbf{x}_m ou \mathbf{x} . Ce plan est choisi perpendiculaire à l'axe optique. La distance qui le sépare du foyer image principal est arbitraire. Dans la cas d'une modélisation sténopé, il est situé à distance focale du centre de projection

Le modèle sténopé



Le sténopé, à l'origine, c'est un petit trou (le mot vient du grec). C'est sans doute le point de départ de la photographie, puisque c'est la pièce maîtresse de la chambre noire (ou *camera obscura*) qui permet la formation d'une image de la réalité extérieure. Les rayons lumineux passent par ce petit trou et se projettent sur le fond d'une chambre noire de manière inversée pour former l'image. C'est un système sans lentille. Ce sont les propriétés de diffraction de la lumière qui permettent d'obtenir une image nette si la taille du sténopé est bien choisie par rapport à la distance qui le sépare du plan image. Pour réaliser des images correctes, le sténopé doit être parfaitement rond et d'un diamètre précis. Pour une distance focale f donnée, il existe une valeur optimale théorique du diamètre d pour la longueur d'onde moyenne de la lumière (550nm) :

$$d = 0.03679\sqrt{f}$$

Le sténopé est utilisé dès le Ve siècle avant JC en Chine. On en retrouve également la description dans les écrits d'Aristote puis de Platon qui s'en sert pour asseoir sa théorie de l'original et de la copie. Cette idée suivant laquelle l'image est un appauvrissement du réel est liée à la perte d'une dimension lors du passage du réel à la *photographie*.

Les suivants à avoir utilisé le sténopé sont les astronomes. Alhazan, savant égyptien du début XIe, entraînait dans une tente sténopé pour observer notamment les éclipses de soleil et expliquer le spectre lumineux et les couleurs.

C'est seulement au début du XVIIe siècle que le terme de chambre noire fut créé par Kepler. On trouve la première mention du terme pin-hole (trou d'épingle) en 1856. Les premiers grands photographes vont ensuite s'emparer de ces noms et les enrichir, le sténopé va ainsi devenir Lensless, puis Lochkamera ou Hullkamera.

FIG. 2.1 – Petite histoire du sténopé

et devant la caméra. L'intersection de l'axe optique avec le plan image \mathbf{x}_c est le *point principal* (2.2).

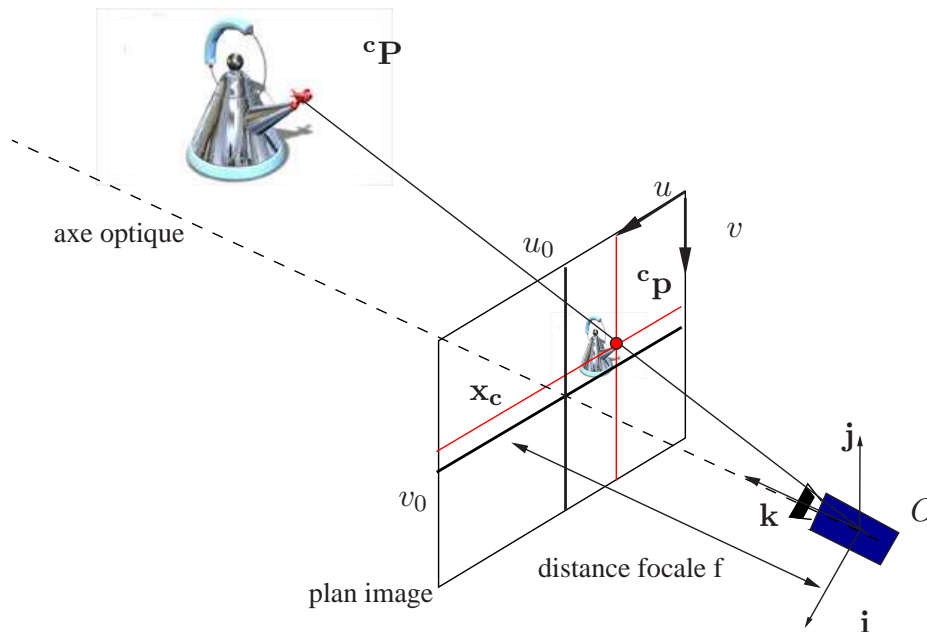


FIG. 2.2 – Modelé sténopé ou trou d'épingle de la caméra

L'origine du repère attaché à une caméra est le centre de projection. Les axes \mathbf{i} et \mathbf{j} sont les axes parallèles au plan image correspondant aux directions des lignes et des colonnes des éléments de l'image. \mathbf{k} est orienté vers le plan image et donc vers la scène.

Soit ${}^c\mathbf{X}({}^cX, {}^cY, {}^cZ, 1)$, un point de l'espace 3D dont les coordonnées métriques sont exprimées dans le repère de la caméra. Sa projection sur le plan image est le point \mathbf{x}_m de coordonnées métriques $(x, y, 1)$ (voir la figure 2.3) :

$$\begin{aligned} x &= f \frac{{}^cX}{{}^cZ} \\ y &= f \frac{{}^cY}{{}^cZ} \end{aligned} \quad (2.6)$$

où f est la distance focale.

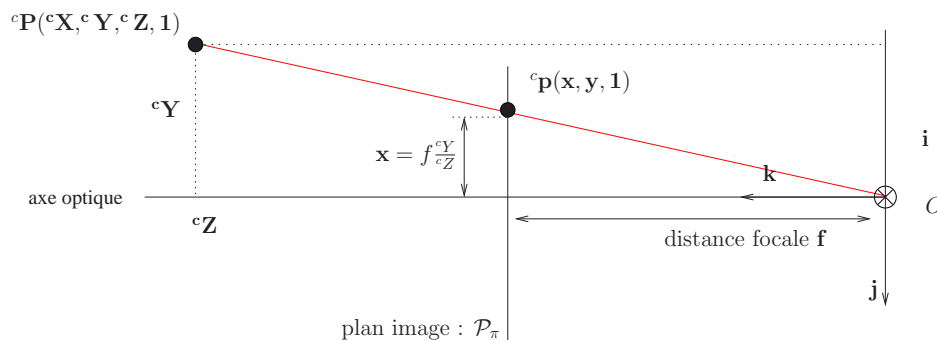


FIG. 2.3 – Projection centrale et notations

Cette projection peut être exprimée sous forme matricielle en formulant le passage par une transformation projective \mathbf{A} de l'espace projectif de dimension 3, \mathbb{P}^3 , vers l'espace projectif de dimension 2 \mathbb{P}^2 .

$$\begin{pmatrix} f^c X \\ f^c Y \\ cZ \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} cX \\ cY \\ cZ \\ 1 \end{pmatrix} \quad (2.7)$$

ou bien dans une forme plus concise :

$$\mathbf{x}_m = \mathbf{A}_p^c \mathbf{X} \text{ avec } \mathbf{A}_p = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.8)$$

2.1.2.2 Les paramètres intrinsèques

L'image numérique est une forme discrétisée de la projection continue du monde 3D sur le plan image. Le plan image est échantillonné suivant un pavage qui est, la plupart du temps, rectangulaire et qui peut parfois être hexagonal ou triangulaire. Chacune des cellules du pavage est caractérisée par sa position dans le pavage et la quantité de lumière qu'elle reçoit.

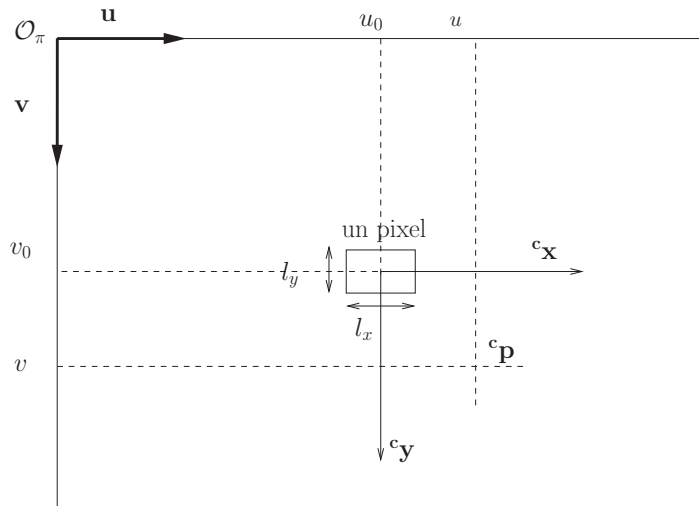


FIG. 2.4 – Passage des coordonnées métriques aux coordonnées pixeliques

Les coordonnées d'un élément de l'image, ou *pixel*, ne sont pas directement données en unités métriques mais représentent la position de l'élément sur le pavage qui compose le capteur CCD. Elles sont appelées *coordonnées pixeliques*. Un repère 2D attaché à la grille de pixels est ajouté à la représentation de la caméra, $\mathcal{R}_\pi(\mathcal{O}_\pi, \mathbf{u}, \mathbf{v})$. L'origine des coordonnées pixeliques est rarement le point principal (voir figure 2.4), elle est généralement placée en haut et à gauche de l'image. \mathbf{u} est colinéaire aux lignes et \mathbf{v} aux colonnes.

Ainsi, les paramètres intrinsèques qui permettent de modéliser la caméra sont :

- la dimension des éléments de l'image l_x et l_y ;
- la *distance focale* f ;
- les coordonnées pixeliques du *point principal* u_0 et v_0 , c'est-à-dire l'intersection de l'axe optique avec le plan image

Le passage des coordonnées normalisées $\mathbf{x}_m(x, y, 1)$, exprimées en unités métriques dans le repère de la caméra $\mathcal{R}_c(\mathcal{O}_c, \mathbf{i}, \mathbf{j}, \mathbf{k})$, aux coordonnées pixeliques $\mathbf{x}_p(u, v, 1)$, exprimées dans le repère de l'image $\mathcal{R}_\pi(\mathcal{O}_\pi, \mathbf{u}, \mathbf{v})$, prend en compte le changement d'unités mais aussi un changement d'origine.

$$\begin{aligned} u &= u_0 + \frac{x}{l_x} \\ v &= v_0 + \frac{y}{l_y} \end{aligned} \quad (2.9)$$

L'ensemble de ces paramètres peut être codé globalement sous la forme d'une matrice de projection perspective \mathbf{K}_p . L'équation (2.9), peut s'écrire sous la forme matricielle suivante :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} \frac{1}{l_x} & 0 & u_0 \\ 0 & \frac{1}{l_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.10)$$

et dans une forme plus concise :

$$\mathbf{x}_p = \mathbf{K}_p \mathbf{x} \text{ avec } \mathbf{K}_p = \begin{bmatrix} \frac{1}{l_x} & 0 & u_0 \\ 0 & \frac{1}{l_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

Pour résumer, d'après les équations (2.7) et (2.10), un point 3D dont les coordonnées sont exprimées en unité métrique dans le repère de la caméra aura pour coordonnées pixeliques :

$$\mathbf{x}_p = \mathbf{K}_p \mathbf{A}_p \mathbf{X} \quad (2.12)$$

Cette expression est fréquemment remplacée par une expression équivalente obtenue en déplaçant f de \mathbf{A}_p vers \mathbf{K}_p qui deviennent \mathbf{A} et \mathbf{K} :

$$\mathbf{x}_p = \mathbf{K} \mathbf{A} \mathbf{X} \text{ avec } \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ et } \mathbf{K} = \begin{bmatrix} \frac{f}{l_x} & 0 & u_0 \\ 0 & \frac{f}{l_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

Les paramètres intrinsèques sont donc réduits à 4 paramètres : deux paramètres de changement d'échelle $p_x = \frac{f}{l_x}$, $p_y = \frac{f}{l_y}$ et les coordonnées pixeliques du point principal \mathbf{x}_c u_0 et v_0 . La matrice \mathbf{K} est couramment appelée *matrice d'étalonnage* ou matrice des paramètres intrinsèques de la caméra. Ainsi, sans perte de généralité, toute caméra peut être assimilée à une caméra équivalente de focale égale à 1 et dont les pixels seraient de tailles $p_x = \frac{f}{l_x}$ et $p_y = \frac{f}{l_y}$.

En général, les points de l'espace 3D ne sont pas directement exprimés dans le repère de la caméra mais dans un repère différent, généralement le repère de référence \mathcal{R}_b ou dans le repère attaché à l'objet \mathcal{R}_o . La transformation entre le repère de la caméra et un autre repère d'expression est appelée la *pose* de la caméra. Dans le reste de ce document, elle sera exprimée par une matrice homogène.

Il est nécessaire d'exprimer les coordonnées d'un point 3D \mathcal{X} dans le repère d'une caméra pour pouvoir calculer les coordonnées de sa projection sur le plan image de la caméra. Si les coordonnées métriques d'un point \mathbf{x}_m sont connues, il est également nécessaire de connaître la pose de la caméra pour reprojeter ce point en 3D. La pose de la caméra est essentielle à l'utilisation de la caméra comme un système de mesure, en particulier dans les tâches de reconstruction 3D et de localisation.

Soit ${}^o\mathbf{X}({}^oX, {}^oY, {}^oZ, 1)$, un point défini dans le repère \mathcal{R}_o . Connaissant la matrice ${}^c\mathbf{M}_o$, pour exprimer ce point dans le repère de la caméra \mathcal{R}_c , il suffit d'écrire :

$${}^c\mathbf{X} = {}^c\mathbf{M}_o {}^o\mathbf{X} \quad (2.14)$$

Finalement, les coordonnées pixeliques d'un point défini dans le repère \mathcal{R}_o sont trouvées en utilisant les équations (2.14) et (2.13) :

$${}^c\mathbf{x}_p = \mathbf{K}\mathbf{A}{}^c\mathbf{M}_o{}^o\mathbf{X} \quad (2.15)$$

$${}^c\mathbf{x}_p = \mathbf{P}{}^o\mathbf{X} \quad (2.16)$$

où \mathbf{P} est une matrice de dimension 3×4 appelée *matrice de projection de la caméra*. Dans la suite de ce manuscrit, nous utiliserons cette matrice pour effectuer la projection d'un point sur un plan image.

Le modèle sténopé utilisé en vision par ordinateur n'est pas exact pour une optique à lentille et des déformations apparaissent qui sont essentiellement dues à la forte courbure aux bords des lentilles de l'objectif. Plus la distance focale est faible et plus ces effets sont significatifs. La déformation la plus importante est une distorsion radiale qui apparaît lors de la projection du monde 3D sur le plan image 2D de la caméra. Si cette déformation peut être corrigée, alors la caméra se comporte comme un système parfaitement linéaire. Une description de ces modèles peut être trouvée dans [Faugeras 93, Hartley 01]. Pour savoir quels sont les coordonnées pixeliques d'un point 3D, il faut d'abord le projeter dans le plan image pour obtenir ses coordonnées métriques idéale, puis calcule ses coordonnées métrique réelles en tenant compte de la distorsion et enfin les convertir en coordonnées pixeliques. La transformation permettant de passer des coordonnées idéales aux coordonnées réelles peut être évalué au cours d'une étape d'étalonnage de la caméra. La technique d'étalonnage que nous avons utilisée repose sur une méthode de minimisation de type Levenberg Marquardt [Chaumette 90a].

Hypothèse 8 *Notre système de vision est étalonné*

Dans le reste de ce document, le système de vision est supposé étalonné, c'est-à-dire l'ensemble des paramètres intrinsèques $\{p_x, p_y, u_0, v_0, L\}$ est connu et sa pose par rapport à un repère de référence \mathcal{R}_b peut être mesurée à chaque instant. Ainsi, le système de vision pourra être utilisé comme un système de mesure.

2.1.2.3 **Rétro-projection d'un point.**

L'équation (2.12) permet de projeter un point 3D ${}^c\mathbf{X}$ sur le plan image d'une caméra. Si les coordonnées d'un point 2D \mathbf{x}_p sont connus, quelle information 3D peut-on en déduire ?

D'après l'équation (2.12), l'ensemble des points ${}^c\mathbf{X}$ qui se projettent sur \mathbf{x}_p est défini par

$$\begin{aligned} {}^c\mathbf{X} &= \mathbf{A}^{-1}\mathbf{K}^{-1}\mathbf{x}_p \\ {}^c\mathbf{X} &= \mathbf{A}^{-1}\mathbf{x}_m \end{aligned}$$

qui équivaut à

$$\begin{cases} {}^cX - {}^cZx = 0 \\ {}^cY - {}^cZy = 0 \end{cases} \quad (2.17)$$

Cette équation définit deux plans dans l'espace euclidien, c'est-à-dire les paramètres d'une droite 3D. Cette droite passe par le centre de projection et le point $\mathbf{x}_m(x, y, 1)^\top$, elle forme la rétro-projection 3D (*back-projection*) du point 2D.

La position du point ${}^c\mathbf{X}$ sur cette droite ne peut évidemment pas être déterminée à partir de l'équation (2.17). L'information donnée par une seule image est donc insuffisante pour calculer sa position. Au moins deux vues d'un même point sont nécessaires à sa localisation.

Une seconde vue peut être obtenue en utilisant un banc de stéréovision, comme le propose le travail antérieur à cette thèse [Leroux 06], c'est-à-dire deux caméras montées sur un même support rigide. Cependant, pour obtenir une localisation précise, il est préférable d'utiliser des vues relativement éloignées qui peuvent être obtenues simultanément par plusieurs caméras ou successivement par une

même caméra, par exemple, comme proposé par l'algorithme du SLAM (*Simultaneous Localisation And Mapping*) [Davison 03] ou dans les algorithmes de *structure from motion*.

Si la position relative des caméras peut être estimée ou mesurée, un lien géométrique peut être établi entre les projections du monde sur les plans images des deux caméras. Ce lien est la géométrie épipolaire qui est présentée dans le paragraphe suivant.

2.1.3 Système à deux caméras

Le système de vision permet d'acquérir simultanément deux images issues de chacune des caméras, déportée et embarquée.

La figure 2.5 présente la géométrie du système à deux vues (1 et 2), ou géométrie épipolaire, qui relie les structures 3D de la scène à leur projection dans les plans images au travers de la description des matrices essentielles ou fondamentales, c'est-à-dire dans l'espace normalisé ou pixelique.

Dès lors que la position relative 1M_2 des repères liés aux caméras est connue, la géométrie épipolaire peut être calculée. Sinon, elle peut être estimée à partir de l'analyse de zones communes aux images acquises par les caméras.

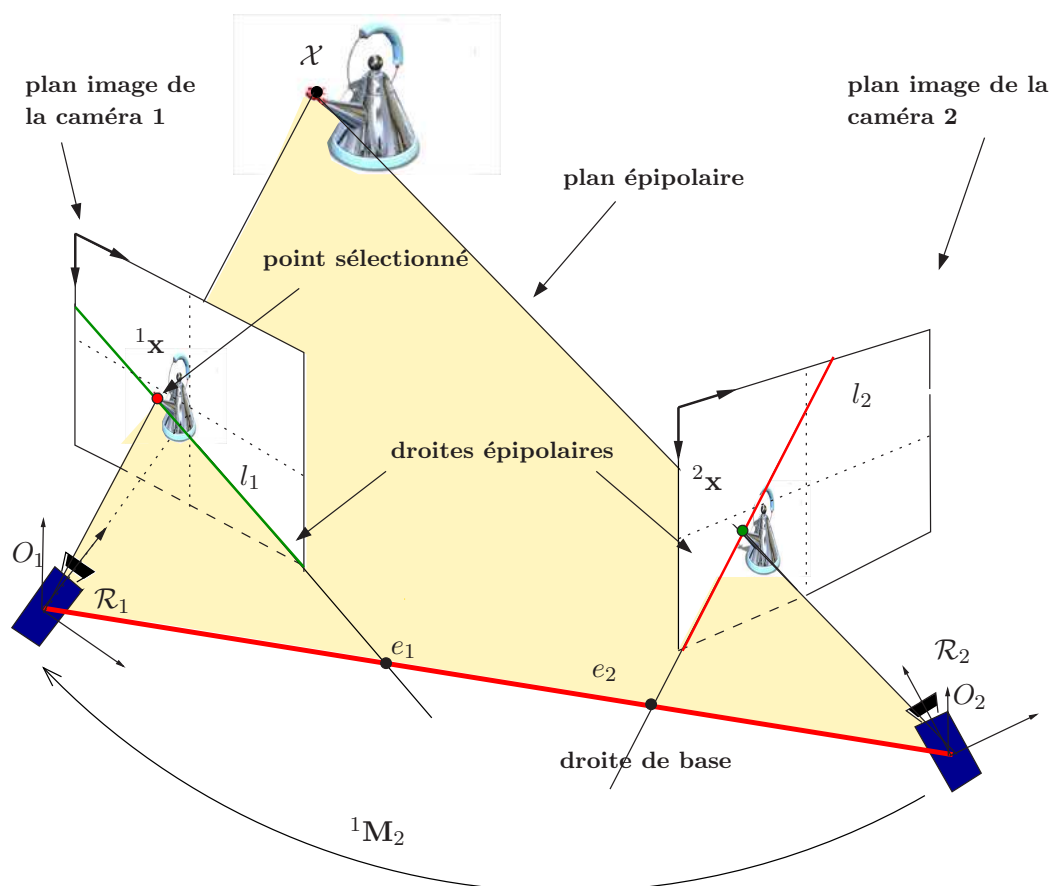


FIG. 2.5 – La géométrie épipolaire de deux vues. Le couple de points homologues est 1x_m et 2x_m . Les centres des caméras sont O_1 et O_2 . Les épipoles sont e_1 and e_2 dans le plan image de la caméra 1 et dans celui de la caméra 2, respectivement. La droite l_1 est la droite épipolaire associée à 2x_m . La droite l_2 est la droite épipolaire associée à 1x_m .

2.1.3.1 Aspects géométriques

Géométriquement, un point 3D \mathcal{X} se projette dans le plan image des deux caméras en deux points 2D ${}^1\mathbf{x}$ et ${}^2\mathbf{x}$. Ces deux points sont dits *homologues*.

À tout point ${}^1\mathbf{x}$ du plan image de la première caméra, il est possible d'associer une droite passant par ce point et le centre optique de la caméra \mathcal{O}_1 . Cette droite se projette dans le plan image de la seconde caméra en une droite l_2 qui est le lieu géométrique des points potentiellement homologues au point cliqué (cf. figure 2.5).

Ainsi, la recherche du point ${}^2\mathbf{x}$ homologue à ${}^1\mathbf{x}$ se limite à l'examen de la droite l_2 dans l'image de la caméra déportée qui est la projection de la ligne de vue passant par le point cliqué. Cette droite est la *droite épipolaire* associée au point cliqué et la géométrie associée aux deux vues est nommée *géométrie épipolaire*.

La droite qui joint les deux centres optiques est appelée la *droite de base*. Les points d'intersection de la droite de base avec chacun des plans images sont les *époles*, \mathbf{e}_1 et \mathbf{e}_2 . Ils représentent la projection centrale du centre optique de l'une des caméras dans le plan image de l'autre caméra. Toutes les droites épipolaires passent par les époles, formant un faisceau de droites épipolaires.

Le plan formé par les centres optiques \mathcal{O}_1 et \mathcal{O}_2 et le point 3D \mathcal{X} est le *plan épipolaire* Π (cf. figure 2.5).

2.1.3.2 Aspects algébriques

Pour alléger les notations, les coordonnées homogènes métriques d'un point 2D dans le plan image de la caméra a seront notées ${}^a\mathbf{x}$, dans la suite de ce paragraphe.

Les points ${}^1\mathbf{x}$ et ${}^2\mathbf{x}$ appartiennent par définition aux droites $(\mathcal{O}_1, {}^1\mathcal{X})$ et $(\mathcal{O}_2, {}^2\mathcal{X})$. Les deux points homologues et les centres optiques sont donc coplanaires, leur produit mixte est donc nul :

$$\begin{aligned} {}^1\mathbf{x}^\top ({}^1\mathbf{t}_2 \times {}^d\mathbf{R}_2 {}^2\mathbf{x}) &= 0 \\ {}^1\mathbf{x}^\top [{}^1\mathbf{t}_2]_{\times} {}^1\mathbf{R}_2 {}^2\mathbf{x} &= 0 \end{aligned}$$

avec ${}^1\mathbf{t}_2$ et ${}^1\mathbf{R}_2$ les composantes de la transformation de la caméra déportée à la caméra embarquée et $[\mathbf{a}]_{\times}$ est une matrice antisymétrique qui permet d'écrire le produit vectoriel de deux vecteur sous la forme d'une multiplication matricielle.

Et finalement, la contrainte épipolaire s'écrit :

$${}^1\mathbf{x}^\top {}^1\mathbf{E}_2 {}^2\mathbf{x} = 0 \quad (2.18)$$

Algébriquement, la relation liant les points \mathbf{x} d'une image aux droites épipolaires de l'autre image est la *contrainte épipolaire* :

$$\begin{aligned} {}^1\mathbf{x}^\top {}^1\mathbf{E}_2 {}^2\mathbf{x} &= 0 \\ {}^2\mathbf{x}^\top {}^1\mathbf{E}_2 {}^1\mathbf{x} &= 0 \\ {}^2\mathbf{x}^\top {}^2\mathbf{E}_1 {}^1\mathbf{x} &= 0 \end{aligned}$$

où ${}^1\mathbf{E}_2$ est la *matrice essentielle*. Le système étant étalonné, cette matrice peut être déduite de la position relative courante des deux caméras ${}^1\mathbf{M}_2$.

La matrice essentielle décrit la relation entre la contrainte épipolaire et les paramètres extrinsèques du système de vision. En effet, $l_2 = {}^2\mathbf{E}_1 {}^1\mathbf{x}_m$ est une droite dans la vue de la caméra embarquée. L'égalité ${}^2\mathbf{x}^\top l_2 = 0$ impose que le point ${}^2\mathbf{x}$ se trouve sur la droite. Cette droite correspond à la projection de la droite associée à ${}^1\mathbf{x}_m$ dans le plan image de la caméra embarquée.

Pour intégrer les paramètres intrinsèques des caméras dans la contrainte épipolaire, il faut passer par la *matrice fondamentale* ${}^1\mathbf{E}_2$ qui permet d'utiliser directement les coordonnées pixeliques. Soit \mathbf{x}_p

un point exprimé en coordonnées pixeliques et \mathbf{x}_m le même point noté en *coordonnées normalisées*. Comme l'a montré le paragraphe précédent, la relation $\mathbf{x}_p = \mathbf{K}\mathbf{x}_m$, avec \mathbf{K} la matrice d'étalonnage de la caméra, permet le passage d'un espace de coordonnées à l'autre.

$$l_2 = {}^2\mathbf{F}_1 {}^2\mathbf{x}_p \quad (2.19)$$

La relation entre la matrice essentielle et la matrice fondamentale est :

$${}^2\mathbf{E}_1 = \mathbf{K}_2^\top {}^2\mathbf{F}_1 \mathbf{K}_1 \quad (2.20)$$

où K_1 et K_2 sont les matrice d'étalonnages, respectivement de la caméra 1 et de la caméra 2.

Finalement, la contrainte épipolaire devient :

$${}^1\mathbf{x}_p^\top {}^1\mathbf{F}_2 {}^2\mathbf{x}_p = 0 \quad (2.21)$$

La géométrie épipolaire correspond à la géométrie du système de vision. Elle est indépendante de la structure de la scène et ne dépend que des paramètres intrinsèques des caméras et de leurs positions relatives. Quelle que soit la scène considérée et connaissant l'étalonnage du système de vision, peut être associé au point cliqué le lieu géométrique de son homologue potentiel dans la vue de la caméra embarquée. Ce lieu géométrique est la projection de la droite passant par le point cliqué et le centre optique.

2.2 Un outil de saisie guidé par la vision

D'après les hypothèses énoncées en fin du premier chapitre, voir l'annexe C, la seule information disponible est un *clik* correspondant à la position approximative de l'objet dans la vue de la caméra déportée.

Dans un premier temps, les repères et les notations permettant de décrire notre système seront présentés. Puis, en utilisant les fondements théoriques décrits dans le paragraphe à venir, l'information qui peut être déduite du *clik* sera présentée.

2.2.1 Repères et Notations

Un certain nombre de repères nécessitent d'être définis une fois pour toutes afin d'indexer les différentes parties du système robotique (voir figure 2.6). Tous les repères positionnés sont des repères cartésiens orthonormés directs de la forme $\mathcal{R}(\mathcal{O}, \mathbf{i}, \mathbf{j}, \mathbf{k})$.

La figure 2.6 présente également les transformations entre les différents repères qui peuvent être exprimées par des matrices homogènes. Ce paragraphe définit les repères et les transformations associées en précisant si celles-ci sont inconnues, estimées ou calculées.

Notons \mathcal{R}_w , le repère absolu, *world frame* et \mathcal{R}_b le repère attaché à la base du bras. Même si la base peut être mobile, cette étude s'intéresse à la phase finale de saisie durant laquelle l'objet est dans la zone de travail du bras. La transformation ${}^w\mathbf{M}_b$ est constante mais inconnue. Dans le reste du document, le repère de référence sera \mathcal{R}_b .

Le repère \mathcal{R}_g , *gripper frame*, est attaché à la pince, avec pour origine le point d'attache de la pince, \mathbf{k} dans l'axe de la pince, \mathbf{j} perpendiculaire aux mors et dans le plan des mors. La position et l'orientation de la pince, à laquelle le repère \mathcal{R}_g est attaché, sont exprimées dans le repère \mathcal{R}_b par la transformation ${}^b\mathbf{M}_g$. Elles varient au cours du temps et elles peuvent être calculées, à chaque instant, à partir de la lecture de la position angulaire des articulations du bras en utilisant son modèle géométrique direct (voir annexe B).

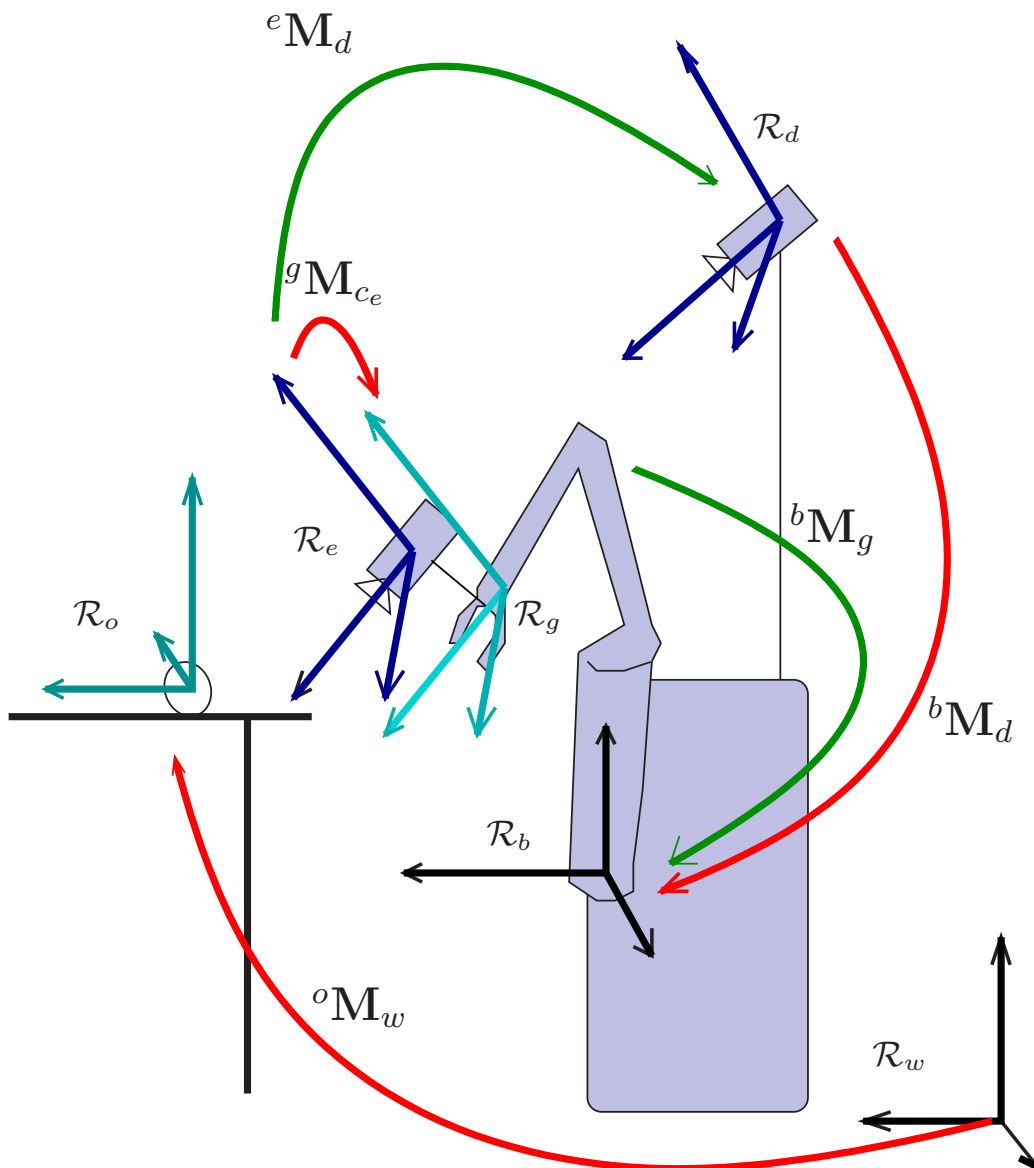


FIG. 2.6 – Définition des repères

Les repères \mathcal{R}_e et \mathcal{R}_d sont respectivement attachés à la caméra embarquée et à la caméra déportée. L'origine d'un repère lié à une caméra est son centre optique et le vecteur directeur \mathbf{k} est collinéaire à l'axe optique. Les transformations liant la caméra embarquée à la pince ${}^b\mathbf{M}_e$ et la caméra déportée à la base du bras ${}^b\mathbf{M}_d$ sont constantes et peuvent être calculées lors d'une étape d'étalonnage [Tsai 89].

Un dernier repère est attaché à l'objet d'intérêt \mathcal{R}_o . L'objet de ce chapitre est de présenter une méthode d'estimation de la position de \mathcal{R}_o dans le repère de référence \mathcal{R}_b pour finalement positionner la caméra embarquée \mathcal{R}_e de sorte à ce que l'objet soit inclus dans son champ de vision. D'après les hypothèses listées dans la première partie, la scène est statique et la position de l'objet à saisir ${}^b\mathbf{M}_o$ est donc constante. Elle est inconnue *a priori*. La pose de l'objet dans le repère \mathcal{R}_e , ${}^o\mathbf{M}_e$, ou \mathcal{R}_d , ${}^o\mathbf{M}_d$, pourrait être calculée à partir des informations visuelles données par l'une ou l'autre des caméras si un modèle 3D de l'objet [Comport 05] ou une base d'apparences était donné [Lowe 04].

2.2.2 Stratégie de localisation.

D'après l'équation (2.17), connaissant les coordonnées pixeliques d'un *clic* dans l'image de la caméra déportée étalonnée, une ligne de vue qui porte l'objet peut être calculée (voir figure 2.7).

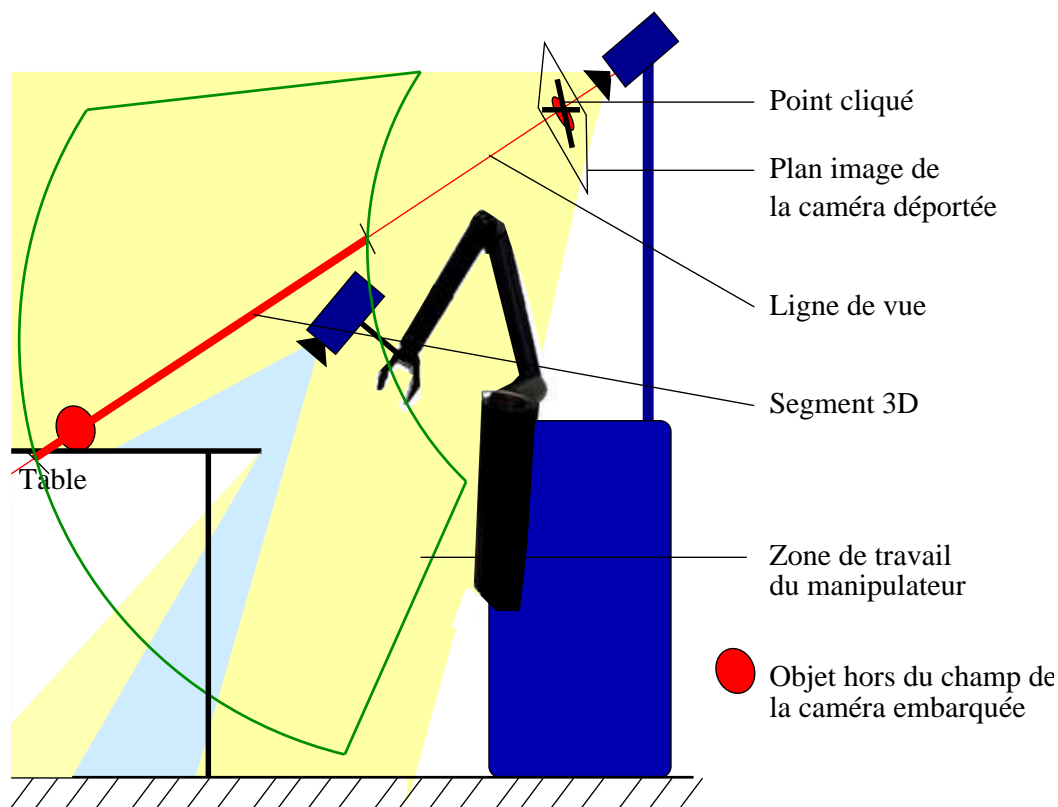


FIG. 2.7 – Définition de la ligne de vue, issue du point cliqué par l'utilisateur

L'objet se trouve dans la zone de travail du bras (voir les hypothèses du chapitre 1), sa profondeur sur la ligne de vue peut donc être bornée par les limites de la zone de travail du bras. L'objet se situe sur la ligne de vue associée au *clic* entre les profondeurs D_{min} et D_{max} . Au moins deux vues sont nécessaires pour déterminer sa localisation sur la ligne de vue.

Dans le cadre de notre étude, une seconde vue est donnée par la caméra embarquée, dont les 6 degrés de liberté peuvent être commandés, dans la zone de travail du bras. La position relative des caméras ${}^d\mathbf{M}_e$ est mesurée et le système de vision est étalonné. La contrainte épipolaire peut donc être calculée (voir équation 2.18) ainsi que les paramètres de la ligne épipolaire associée au *clac*, projection de la ligne de vue dans la vue de la caméra embarquée :

$${}^e l = {}^e \mathbf{F}_d {}^d \mathbf{x}_{clac} \quad (2.22)$$

La stratégie de localisation (voir figure 2.8) consiste à déplacer la caméra embarquée pour examiner la ligne épipolaire ${}^e l$ de D_{min} jusqu'à D_{max} . L'objet n'est pas visible à chaque instant dans l'image de la caméra embarquée. Cependant il se projette nécessairement dans quelques images qu'il faudra identifier. Au fur et à mesure du parcours, les informations issues des images seront traitées pour repérer l'objet et en déduire une estimation de sa position.

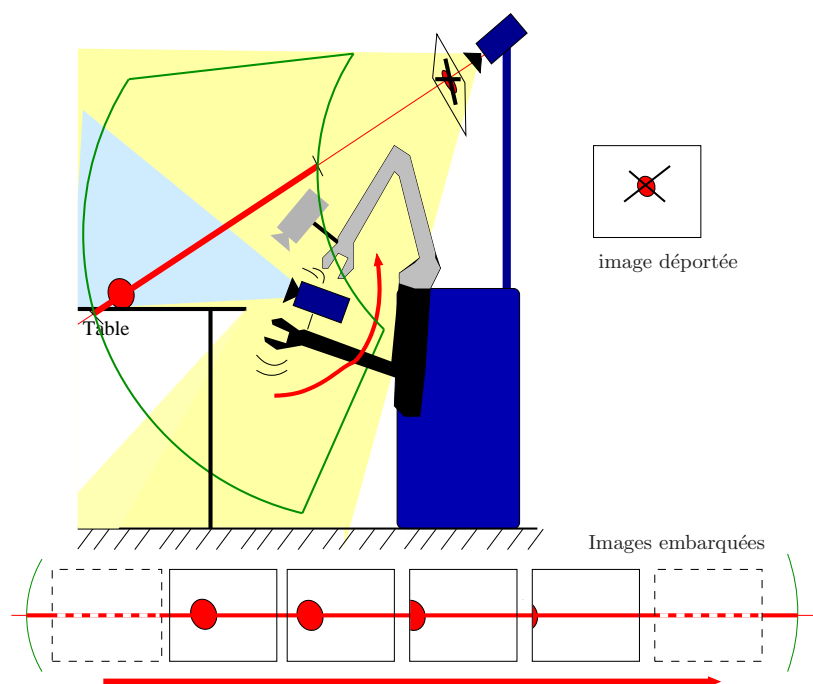


FIG. 2.8 – Stratégie de localisation : la caméra embarquée est déplacée de sorte à acquérir des vues du segment 3D qui porte l'objet d'intérêt.

Le prochain chapitre décrit la loi de commande qui assure le parcours de la ligne de vue et le chapitre suivant présentera les traitements qui permettent de repérer l'objet dans les images.

Une commande visuelle basée sur la géométrie épipolaire

Ce chapitre présente la loi de commande qui est appliquée à la caméra embarquée pour localiser l'objet à saisir. Elle consiste à parcourir la ligne de vue associée au *clac*, en parcourant en fait la ligne épipolaire associée au *clac*, maintenue centrée et horizontale dans l'image embarquée. La commande du mouvement de la caméra repose sur des informations 2D, les paramètres de la ligne épipolaire, et consiste à réaliser une tâche de centrage de la ligne épipolaire et une tâche de parcours de cette ligne.

L'avantage d'utiliser une commande bouclée sur la position de la ligne épipolaire est qu'elle permet de relâcher l'hypothèse d'un système fixe en s'adaptant à des petits mouvements relatifs de l'objet et de la caméra déportée, simplement en suivant le voisinage du *clac*. Par exemple, cela permet de compenser des petits mouvements d'un fauteuil roulant au cours de l'asservissement.

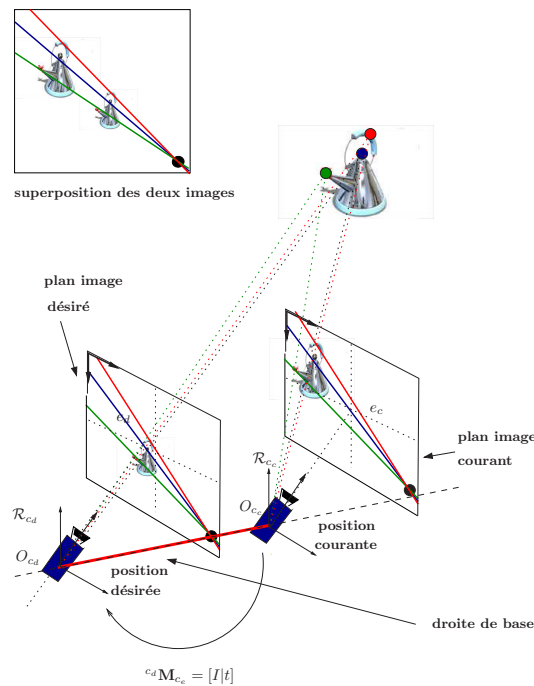
Des travaux se sont intéressés à des commandes basées sur la géométrie épipolaire, essentiellement pour des applications de retour à leur position initiale, en anglais *homing*. Des commandes ont été développées avec succès pour des bras manipulateurs [Rives 00, Basri 99, Piazzi 03, Marotta 02] (mouvement en six degrés de liberté) et, ces dernières années, pour des systèmes mobiles holonomes [Chezi 06, Piazzi 03] ou non holonomes en utilisant des caméras sténopé [LopezNicolas 06, Mariottini 07] ou bien encore catadioptriques [Mariottini 05] (dans ce cas, seuls des mouvements planaires des caméras sont autorisés).

Les commandes développées reposent sur des informations 2D virtuelles calculées à partir de l'estimation en ligne de la géométrie épipolaire [Hartley 97], telle la position des épipoles ou la position de droites épipolaires. Seule l'orientation relative des caméras peut être rectifiée en utilisant ces informations. En effet, lorsque les caméras sont alignées, une configuration particulière, dite *auto-épipolaire*, est atteinte. Dans cette configuration, les épipoles et les droites épipolaires sont confondus (voir l'encadré 3.1).

Dans le cadre de notre étude, la géométrie épipolaire peut être calculée à partir de la position relative des deux caméras, écartant les problèmes liés à son estimation en ligne. Contrairement aux méthodes de *homing*, notre objectif n'est pas de rejoindre une position connue par avance mais de parcourir et d'observer une ligne de vue.

Dans une première section, nous présenterons le principe des asservissements visuels 2D puis un formalisme qui permet de réaliser l'asservissement simultané de plusieurs tâches en utilisant les propriétés

Erreur en rotation : Configuration auto-épipolaire



La configuration auto-épipolaire est une configuration remarquable de la géométrie de deux vues qui advient lorsque les deux repères caméra sont alignés. La transformation qui permet de passer d'un repère à l'autre est une translation dans la direction de la ligne de base. Les épipoles et les lignes épipolaires correspondant à deux points homologues ont les mêmes coordonnées dans les deux vues et sont appelés *auto-épipoles*. Réciproquement, si les lignes épipolaires et les épipoles ont la même position dans les deux vues, alors les repères caméra sont alignés.

Un premier algorithme consiste à extraire un ensemble de points dans les images courante et désirée. Une mise en correspondance de ces points permet d'estimer la géométrie épipolaire et ainsi, la position des lignes épipolaires. L'idée est de déplacer la caméra de sorte à annuler la distance séparant un point de la ligne épipolaire associée et ainsi annuler l'erreur en rotation.

Un autre algorithme repose sur l'extraction des bi-tangentes. Une bi-tangente est une droite tangente aux contours d'un même objet lorsque l'image courante et l'image désirée sont superposées. En configuration auto-épipolaire, les bi tangentes passent par l'auto-épipole [Faugeras 93, LonguetHiggins 81]. Cette propriété est utilisée dans [Piazzì 04, Piazzì 03] pour asservir un bras manipulateur et dans [Mariottini 04, Mariottini 05] pour le retour à la base de robots mobiles non holonomes. L'idée de l'algorithme est de faire pivoter la caméra jusqu'à ce que les bi-tangentes soient concourantes en un unique point.

FIG. 3.1 – Les propriétés de la configuration auto épipolaire permettent d'aligner les caméras.

de redondance d'un bras manipulateur à 6 axes. Ensuite, nous présenterons notre loi de commande. Enfin les résultats obtenus sur un robot cartésien (AFMA Robotics) et sur un robot anthropomorphe (RX90) seront présentés.

3.1 Loi de commande par asservissement visuel

Les techniques permettant de réaliser une commande à partir d'informations visuelles sont les techniques d'asservissements visuels. Ce paragraphe en décrit le principe général. Il n'est pas question ici de faire un état de l'art exhaustif du domaine mais de présenter les méthodes qui sont utiles à la résolution du problème de localisation de l'objet. Ce paragraphe s'appuie principalement sur le tutoriel en deux parties, publié dans le *IEEE Robotics and Automation Magazine* [Chaumette 06, Chaumette 07].

L'asservissement visuel d'un système équipé d'une caméra consiste à contrôler ses déplacements pour minimiser l'erreur entre une information visuelle courante $\mathbf{s}(\mathbf{r})$, où \mathbf{r} est la pose courante de la caméra et une information visuelle désirée \mathbf{s}^* , ou consigne. La figure 3.2 présente les deux montages les plus usuels en commande visuelle. Si la caméra est embarquée sur le système à asservir, il s'agit de contrôler ses mouvements pour réguler la tâche, le système est dit *eye in hand*. Si la caméra est fixée sur un support extérieur, la commande permet de déplacer l'effecteur jusqu'à ce qu'il atteigne une position désirée dans l'image. Le terme de *eye to hand* est rattaché à ce type de montage.

Des travaux se sont intéressés à des systèmes mixtes, dit *eye-in-hand/eye-to-hand* [Flandin 00, Elena 03, Muis 05, Lippiello 05, Horaud 06, Lippiello 07]. La plupart du temps, la commande est découplée : typiquement la caméra *eye-in-hand* commande les rotations et la caméra *eye-to-hand* commande les translations [Elena 03, Flandin 00, Muis 05]. Cependant, il peut être intéressant d'utiliser les données issues des deux caméras pour commander l'ensemble des degrés de liberté du système, par exemple, pour tenir compte des occultations dans chacune des vues [Lippiello 05, Lippiello 07].

Dans le cadre de notre étude, le montage est mixte *eye-in-hand/eye-to-hand* mais la commande est une commande *eye-in-hand*. En effet, comme nous le verrons, seuls les déplacements de la caméra embarquée sont commandés. La coopération entre les deux caméras se résume à utiliser le point sélectionné dans la vue déportée pour construire la loi de commande de la caméra embarquée.

3.1.1 Le formalisme de l'asservissement visuel

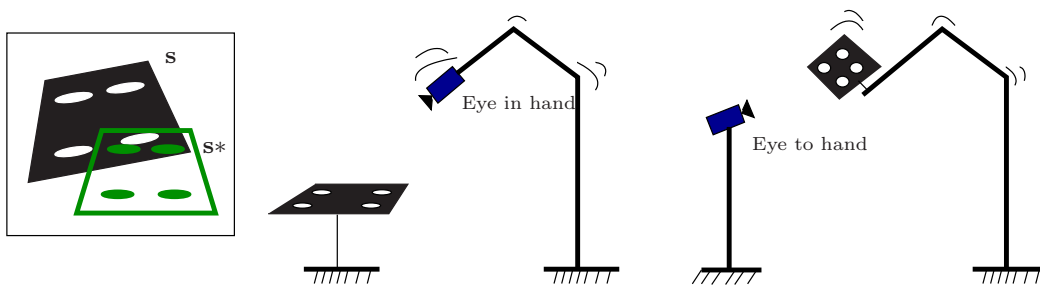


FIG. 3.2 – Dispositif d'asservissement visuel sur 4 points ; à gauche, un montage caméra embarquée, dit *eye-in-hand* ; à droite, un montage caméra déportée, dit *eye-to-hand*.

Pour commander un système de telle sorte que l'information visuelle courante $\mathbf{s}(\mathbf{r})$ se superpose à la consigne \mathbf{s}^* , il est nécessaire de décrire la relation entre les déplacements de ce système et la variation de $\mathbf{s}(\mathbf{r})$. La dérivée de $\mathbf{s}(\mathbf{r})$ par rapport au temps s'écrit :

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v} \quad (3.1)$$

où \mathbf{L}_s est la *matrice d'interaction* associée à $\mathbf{s}(\mathbf{r})$ et $\mathbf{v} = (v, \omega)$ est le torseur cinématique. \mathbf{L}_s dépend de la valeur courante de \mathbf{s} mais aussi de la distance séparant la caméra de l'objet considéré. Cette distance ne peut pas être mesurée à partir de l'information 2D. La matrice d'interaction permet de relier le torseur cinématique de la caméra ou de l'objet, respectivement pour une configuration *eye-to-hand* ou *eye-in-hand*, aux variations des informations visuelles.

Si le système est un bras manipulateur, \mathbf{v} peut être exprimé en considérant la position articulaire de chacun des axes du bras :

$$\mathbf{v} = {}^c\mathbf{V}_g \mathbf{J}(q) \dot{\mathbf{q}} \quad \text{avec } {}^c\mathbf{V}_g = \begin{bmatrix} {}^c\mathbf{R}_g & [{}^c t_g]_{\times} {}^c\mathbf{R}_g \\ 0 & {}^c\mathbf{R}_g \end{bmatrix} \quad (3.2)$$

où

- \mathbf{q} est un vecteur de dimension égale au nombre de degrés de liberté du bras. Ses coordonnées sont les positions de chacune des liaisons du bras. q représente donc la *position articulaire* du bras.
- $\dot{\mathbf{q}} = \frac{\partial \mathbf{q}}{\partial t}$ est la vitesse articulaire.
- $\mathbf{J}(q) = \frac{\partial \mathbf{r}}{\partial \mathbf{q}}$ est le jacobien du robot qui relie des déplacements cartésiens exprimés dans \mathcal{R}_b aux déplacements articulaires.
- ${}^c\mathbf{V}_g$ est la transformation qui permet de passer d'une vitesse exprimée dans le repère de l'effecteur (*gripper*) à une vitesse exprimée dans le repère de la caméra. Elle dépend de la pose de la caméra dans le repère de l'effecteur ${}^g\mathbf{M}_c$ qui est fixe et peut être calculée lors d'une phase d'étalonnage.

L'équation (3.1) devient alors :

$$\dot{\mathbf{s}} = \mathbf{J}_\pi \dot{\mathbf{q}} \quad \text{avec } \mathbf{J}_\pi = \mathbf{L}_s {}^c\mathbf{V}_g \mathbf{J}(q) \quad (3.3)$$

où \mathbf{J}_π est le jacobien image qui permet de relier la variation des positions articulaires à la variation des informations visuelles et dépend de la matrice d'interaction.

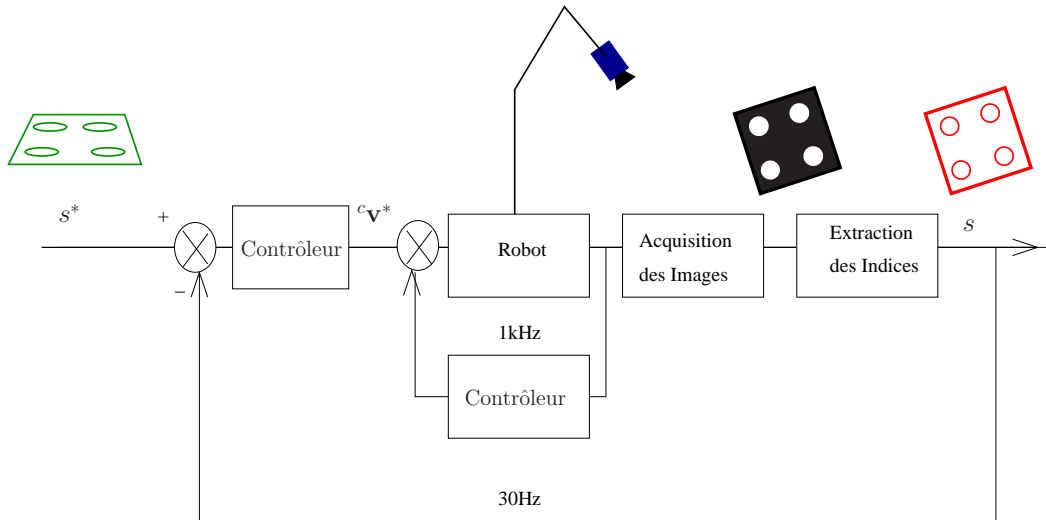


FIG. 3.3 – Schéma bloc simplifié d'un asservissement visuel indirect

La figure 3.3 présente de manière simplifiée la boucle de commande d'un asservissement visuel. L'utilisation du formalisme de la fonction de tâche [Samson 91, Espiau 92] pour réguler l'erreur $\Delta = \mathbf{s}^* - \mathbf{s}$ permet d'écrire :

$$\mathbf{e} = (\mathbf{s}^* - \mathbf{s}(\mathbf{r})) \quad (3.4)$$

Des travaux ont établi des lois de commande qui tiennent compte des modèles dynamiques de la caméra et du système pour réaliser des asservissements visuels rapides (à plus de 50Hz) : lois de commande non linéaires [Hashimoto 93a], optimales de type LQ [Hashimoto 93b] et LQP, ou bien encore utilisant des commandes prédictives, GPC [Gangloff 98] ou des contrôleurs H_∞ .

Dans le cadre de cette étude, les asservissements visuels considérés sont des asservissements indirects dont la fréquence d'échantillonnage est celle de l'acquisition des images de la caméra, c'est-à-dire 25-30Hz. La dynamique du système est contrôlée par les couches de commande bas niveau qui ont une fréquence d'échantillonnage de 1kHz. Au regard du rapport de ces fréquences d'échantillonnage, le système peut donc être considéré comme un intégrateur parfait et une loi de commande proportionnelle permet d'obtenir une décroissance exponentielle de l'erreur :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (3.5)$$

où λ est le gain du correcteur.

Une nouvelle expression de la dérivée de \mathbf{e} peut être obtenue à partir de la dérivée de l'équation (3.4) et de l'équation (3.3) :

$$\dot{\mathbf{e}} = \dot{\mathbf{s}} = \mathbf{L}_s^c \mathbf{V}_g \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (3.6)$$

La loi de commande permettant une décroissance exponentielle s'écrit à partir des équations (3.5) et (3.6) :

$$\dot{\mathbf{q}} = -\lambda (\mathbf{L}_s^c \mathbf{V}_g \mathbf{J}(\mathbf{q}))^+ \mathbf{e} = -\lambda (\mathbf{J}_\pi)^+ \mathbf{e} \quad (3.7)$$

où \mathbf{M}^+ est la pseudo inverse de Moore-Penrose de \mathbf{M} , telle que $\mathbf{M}^+ = (\mathbf{M}^\top \mathbf{M})^{-1} \mathbf{M}^\top$.

Cette loi de commande est une loi idéale. Dans la pratique, \mathbf{L}_s ne peut pas être parfaitement calculée car elle dépend à la fois de \mathbf{s} et de la profondeur séparant la caméra de l'objet 3D qui n'est pas connue. Ainsi, \mathbf{L}_s ne peut être qu'estimée et son estimation sera notée $\widehat{\mathbf{L}}_s$. La loi de commande (3.7) devient alors :

$$\dot{\mathbf{q}} = -\lambda (\widehat{\mathbf{L}}_s^c \mathbf{V}_g \mathbf{J}(\mathbf{q}))^+ \mathbf{e} = -\lambda (\widehat{\mathbf{J}}_\pi)^+ \mathbf{e} \quad (3.8)$$

Si la loi de commande ne contraint pas tous les degrés de liberté du système, les degrés de liberté disponibles peuvent être utilisés pour l'application d'une seconde loi de commande. Cependant, il est possible que la régulation de cette seconde commande perturbe l'exécution de la première si elles ne sont pas découplées. Le paragraphe suivant présente un formalisme permettant d'appliquer plusieurs commandes sur un même système en établissant une hiérarchie entre les commandes et en garantissant que les commandes secondaires ne perturbent pas la commande principale.

3.1.2 De la redondance en asservissement visuel

Lorsque la commande principale ne contraint pas tous les degrés de liberté du système, les degrés de liberté disponibles peuvent être utilisés pour réaliser des tâches supplémentaires. La question est alors de savoir comment utiliser au mieux ces degrés de liberté restants sans perturber la commande principale. Plusieurs solutions ont été proposées pour associer plusieurs commandes, ou *tâches*.

La première consiste à faire une combinaison linéaire entre les tâches [Nelson 93, Nelson 94b, Nelson 94a]. Cependant, les tâches secondaires peuvent alors modifier la tâche principale. Cette solution ne garantit donc pas qu'il n'y ait pas de conflit entre les tâches. Il est même possible que la loi de commande résultante soit telle qu'aucune des tâches ne soit finalement réalisée.

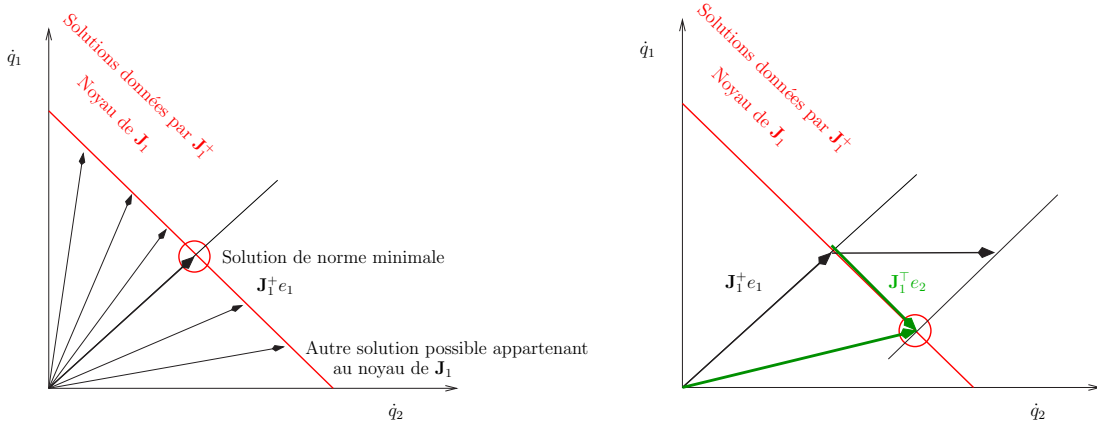


FIG. 3.4 – à gauche : il existe une infinité de solutions, toutes situées sur le noyau \mathbf{J}_1^\top de \mathbf{J}_1 . La pseudo inverse \mathbf{J}_1^+ de \mathbf{J}_1 fournit la solution de norme minimale; à droite : Dans le cas où une tâche secondaire \mathbf{e}_2 est considérée, elle est projetée sur le noyau \mathbf{J}_1^\top de la tâche et n'a donc aucun effet sur la tâche principale.

La seconde solution utilise le principe de la redondance [Siciliano 91] : celui-ci permet d'appliquer des tâches supplémentaires sans perturber la tâche principale. L'idée est de projeter une tâche secondaire sur l'espace des mouvements qui ne perturbent pas la tâche principale. Typiquement, à une tâche principale de centrage peut être associée une tâche d'exploration ou d'évitement de butées. Cela demande d'établir une hiérarchie des tâches. Généralement la tâche d'asservissement visuel est la tâche prioritaire et la tâche d'évitement de butées secondaire. Il est possible d'utiliser ce formalisme pour appliquer plus de deux tâches au système.

Il arrive parfois que la contrainte de la tâche principale soit trop importante pour permettre l'ajout d'une tâche secondaire. Dans ce cas, il faut alterner des phases d'exécution de la commande principale à des phases où les contraintes secondaires s'expriment [Mansard 07].

Le formalisme de l'asservissement visuel à une seule tâche a été présenté précédemment. \mathbf{q} est le vecteur des positions articulaires du manipulateur et $\mathbf{e}_i = \mathbf{s}_i^* - \mathbf{s}_i$ et $i \in \{1, 2\}$ deux fonctions de tâches à réguler. D'après l'équation (3.3), le jacobien relatif à chacune des tâches est défini par :

$$\dot{\mathbf{e}}_i = \dot{\mathbf{s}}_i \quad (3.9)$$

$$= \mathbf{J}_i \dot{\mathbf{q}} \quad (3.10)$$

Si la tâche principale est une tâche d'asservissement visuel et si on note \mathbf{L}_1 la matrice d'interaction associée, alors on peut exprimer le jacobien de la tâche ainsi :

$$\mathbf{J}_1 = -\lambda_1 \mathbf{L}_1^c \mathbf{V}_g \mathbf{J}(\mathbf{q}) \quad (3.11)$$

où λ_1 est le gain du correcteur proportionnel qui assure la décroissance exponentielle de l'erreur liée à la tâche.

La loi de commande qui optimise l'erreur de la tâche principale $\mathbf{e}_1 = \mathbf{s}_1^* - \mathbf{s}_1$ est alors [Espiau 92, Chaumette 06] :

$$\dot{\mathbf{q}} = \widehat{\mathbf{J}}_1^+ \dot{\mathbf{e}}_1 + \mathbf{P}_1 \mathbf{z} \quad (3.12)$$

où $\widehat{\mathbf{J}}_1^+$ est la pseudo inverse d'une approximation ou d'un modèle de \mathbf{J}_1 , \mathbf{z} est un vecteur de contrôle arbitraire et $\mathbf{P}_1 = \mathbf{I} - \widehat{\mathbf{J}}_1^+ \mathbf{J}_1$ (avec \mathbf{I} la matrice identité) est l'opérateur de projection qui garantit que le vecteur de contrôle \mathbf{z} n'affecte pas la réalisation de la tâche principale \mathbf{e}_1 .

En effet quel que soit \mathbf{z} , on a l'égalité $\mathbf{J}_1 \mathbf{P}_1 \mathbf{z} = 0$. En d'autres termes, si \mathbf{J}_1^+ donne une commande de norme minimale, il existe une infinité de solutions permettant d'assurer la régulation [Marchand 04]. Toutes les autres solutions se situent dans le noyau du Jacobien de la tâche. La figure (3.4) illustre la projection de la tâche secondaire sur le noyau de la tâche principale.

Introduisons une deuxième tâche \mathbf{e}_2 et son jacobien \mathbf{J}_2 dans le vecteur \mathbf{z} qui devient :

$$\mathbf{z} = \widehat{\mathbf{J}}_2^+ \mathbf{e}_2 \quad (3.13)$$

Si on inclut (3.13) dans (3.12), la loi de commande permettant de réguler les deux tâches est alors :

$$\mathbf{q} = \widehat{\mathbf{J}}_1^+ \mathbf{e}_1 + \mathbf{P}_1 \widehat{\mathbf{J}}_2^+ \mathbf{e}_2 \quad (3.14)$$

Pour étendre (3.14) à $n \geq 2$ tâches, $\mathbf{e}_1, \dots, \mathbf{e}_n$ dont les jacobiens sont notés $\mathbf{J}_1, \dots, \mathbf{J}_n$, il est tentant d'écrire la loi de commande de la troisième tâche directement à la place de la seconde tâche de l'équation :

$$\dot{\mathbf{q}} = \widehat{\mathbf{J}}_1^+ \mathbf{e}_1 + \mathbf{P}_1 (\widehat{\mathbf{J}}_2^+ \mathbf{e}_2 + \mathbf{P}_2 \widehat{\mathbf{J}}_3^+ \mathbf{e}_3)$$

et en généralisant à n tâches :

$$\dot{\mathbf{q}} = \widehat{\mathbf{J}}_1^+ \mathbf{e}_1 + \mathbf{P}_1 \widehat{\mathbf{J}}_2^+ \mathbf{e}_2 + \mathbf{P}_1 \mathbf{P}_2 \widehat{\mathbf{J}}_3^+ \mathbf{e}_3 + \dots + \left(\prod_{i=1}^{n-1} \mathbf{P}_i \right) \widehat{\mathbf{J}}_n^+ \mathbf{e}_n$$

Cependant, les opérateurs de projection vectorielle \mathbf{P}_i ne sont pas commutatifs ($\mathbf{P}_1 \mathbf{P}_2 \neq \mathbf{P}_2 \mathbf{P}_1$ dans le cas général). Rien ne garantit donc que $\mathbf{P}_1 \mathbf{P}_2 \widehat{\mathbf{J}}_3^+ \mathbf{e}_3$ se projette sur le noyau de \mathbf{e}_2 , c'est-à-dire la tâche \mathbf{e}_2 peut être modifiée par la tâche \mathbf{e}_3 ainsi que par toutes les autres tâches.

Pour que les tâches secondaires n'influencent pas sur la tâche principale et ne se perturbent pas entre elles, il faut garantir que la tâche \mathbf{e}_{i+1} se projette dans les noyaux des i tâches qui lui sont prioritaires, c'est-à-dire dans l'intersection des noyaux des i premières tâches. Soit $\mathbf{N}_{1..i} = \bigcap_{p=1}^i \ker \mathbf{J}_p$ et $\mathbf{J}_i^{\mathbf{A}}$ le jacobien des p premières tâches :

$$\mathbf{J}_i^{\mathbf{A}} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_p \end{bmatrix}$$

Le projecteur vectoriel $\mathbf{P}_i^{\mathbf{A}}$ est le projecteur dans le noyau de $\mathbf{J}_i^{\mathbf{A}}$:

$$\mathbf{P}_i^{\mathbf{A}} = \mathbf{I} - \mathbf{J}_i^{\mathbf{A}+} \mathbf{J}_i^{\mathbf{A}}$$

L'expression d'une loi de commande à plus de deux tâches s'écrit finalement :

$$\dot{\mathbf{q}} = \widehat{\mathbf{J}}_1^+ \mathbf{e}_1 + \mathbf{P}_1^{\mathbf{A}} \widehat{\mathbf{J}}_2^+ \mathbf{e}_2 + \mathbf{P}_2^{\mathbf{A}} \widehat{\mathbf{J}}_3^+ \mathbf{e}_3 + \dots + \mathbf{P}_{n-1}^{\mathbf{A}} \widehat{\mathbf{J}}_n^+ \mathbf{e}_n \quad (3.15)$$

Le lecteur intéressé pourra se référer à l'étude de la mise en œuvre par pile de tâches des lois de commande à plus de deux tâches, dans la thèse de N. Mansard [Mansard 07].

3.2 Loi de commande appliquée à la caméra embarquée pour localiser l'objet

Dans le cas de notre étude, la spécification des tâches à effectuer est la suivante, par ordre de priorité :

1. La droite épipolaire associée à ${}^d \mathbf{x}_m$ doit être amenée au centre du plan image de la caméra mobile ;

2. Tout en maintenant la droite épipolaire horizontale et centrée dans sa vue, la caméra mobile parcourt l'espace de travail du bras à la recherche de l'objet à saisir ;
3. Une troisième tâche d'évitement de butées sera appliquée.

Ce paragraphe présente donc la loi de commande à trois tâches que nous avons développée.

Pour parcourir la ligne de vue, si la tâche principale de centrage est réglée, la tâche secondaire de parcours devrait se limiter à une rotation autour de l'axe perpendiculaire au plan épipolaire, c'est-à-dire l'axe \mathbf{j} du repère de la caméra embarquée [Basri 99]. Cependant, si la régulation de la tâche principale n'est pas parfaite, cette rotation perturbe petit à petit le centrage de la droite épipolaire. L'erreur de la tâche principale augmente et plus rien ne garantit que la droite épipolaire soit dans l'image de la caméra mobile. Le formalisme de la redondance permet de résoudre ce problème en assurant que la tâche secondaire ne perturbe pas la bonne régulation de la tâche principale.

3.2.1 Tâche principale : centrage de la ligne

D'après le paragraphe 2.1.3 consacré à la géométrie épipolaire, la ligne de vue portant le centre optique de la caméra déportée et le point ${}^d\mathbf{x}_m$ se projette dans le plan image de la caméra embarquée sur la droite d'équation :

$$l_e = {}^e\mathbf{E}_d {}^d\mathbf{x}_m \quad (3.16)$$

Pour asservir les déplacements de la caméra à ceux de cette droite dans un plan image, une paramétrisation $\mathbf{s} = (\rho, \theta)$ (voir figure 3.5) est préférée à la représentation non minimale ($ax + by + c = 0$) pour représenter la ligne de vue [Chaumette 90b] :

$$x \cos \theta + y \sin \theta - \rho = 0 \quad (3.17)$$

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

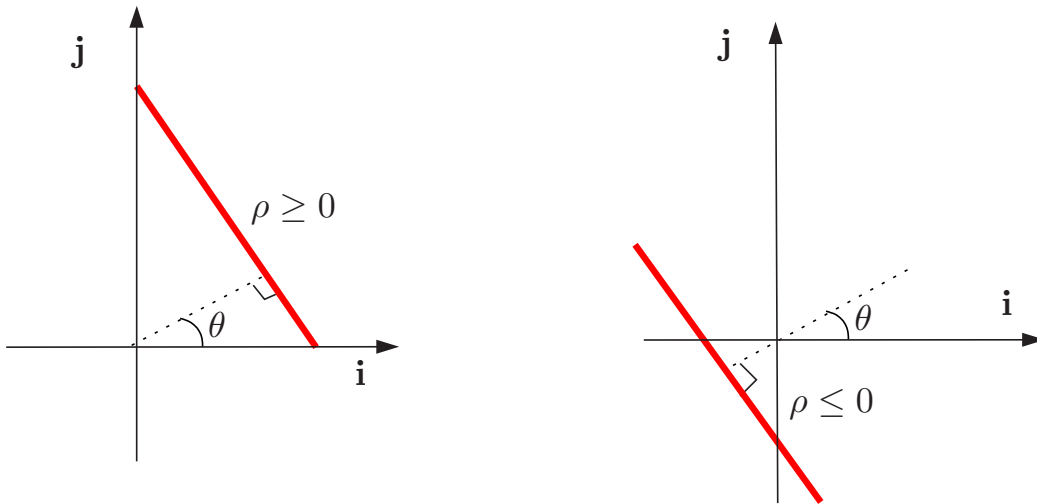


FIG. 3.5 – Représentation des droites 2D $\mathbf{s} = (\rho, \theta)$

Remarque Il faut cependant noter que la représentation (ρ, θ) présente une ambiguïté : la même droite peut être paramétrée indifféremment par $(\rho, \theta \pm 2k\pi)$ et $(-\rho, \theta \pm (2k+1)\pi)$. Pour lever l'ambiguïté, il suffit de limiter l'angle θ entre $-\pi$ et π , le signe de ρ est ainsi fixé.

La matrice d'interaction associée à ${}^e l$ est la suivante [Espiau 92] :

$$\mathbf{L}_1 = \begin{pmatrix} \lambda_\rho \cos\theta & \lambda_\rho \sin\theta & -\lambda_\rho \rho & (1 + \rho^2) \sin\theta & -(1 + \rho^2) \cos\theta & 0 \\ \lambda_\theta \cos\theta & \lambda_\theta \sin\theta & -\lambda_\theta \rho & -\rho \cos\theta & -\rho \sin\theta & -1 \end{pmatrix} \quad (3.18)$$

avec :

$$\begin{cases} \lambda_\rho & = (A\rho \cos\theta + B\rho \sin\theta + C)/D \\ \lambda_\theta & = (A \sin\theta - B \cos\theta)/D \end{cases} \quad (3.19)$$

où $AX + BY + CZ + D = 0$ est un plan, exprimé dans le repère \mathcal{R}_e qui contient la ligne. Pour éviter une configuration dégénérée, où $D = 0$, le plan choisi ne passe pas par le centre optique de la caméra embarquée. On choisit typiquement le plan image de la caméra embarquée.

L'asservissement consiste à centrer horizontalement la ligne \mathbf{s}_1 dans le plan image, d'où $s_l^* = (0, \pi)$. La loi de commande dans l'espace articulaire est donc la suivante :

$$\dot{\mathbf{q}} = -\lambda_1 (\widehat{\mathbf{L}}_1 {}^c \mathbf{V}_g \mathbf{J}(q))^+ (\mathbf{s}_1^* - \mathbf{s}_1) \quad (3.20)$$

Les coordonnées de la ligne épipolaire sont directement calculées à partir de la mesure des positions relatives des deux caméras et de la position du *clic* dans la vue de la caméra déportée (voir équation 3.16). Dans le cas où l'objet est immobile relativement au repère du bras, cet asservissement ne repose donc pas directement sur des informations image mais uniquement sur des informations géométriques, sa désignation en tant "qu'asservissement visuel" est donc en partie abusive. Cependant, ce formalisme permet exactement d'exprimer notre tâche purement géométrique.

Cette tâche ne contraint que deux degrés de liberté. En effet, la matrice jacobienne attachée à la tâche est de rang 2. Il reste donc quatre degrés de liberté qui peuvent être utilisés pour déplacer la caméra embarquée le long de cette ligne de vue afin de rechercher l'objet.

3.2.2 Tâche secondaire : parcours de la ligne

La recherche de l'objet est limitée à un segment 3D qui est l'intersection entre la droite épipolaire et l'espace de travail du robot. Les deux extrémités de ce segment sont des points 3D virtuels ${}^c \mathbf{X}_1 = (X_1, Y_1, Z_1)$ et ${}^c \mathbf{X}_2 = (X_2, Y_2, Z_2)$, exprimés dans le repère \mathcal{R}_d . Ils se projettent dans le plan image de la caméra mobile en deux points $\mathbf{x}_1 = (x_1, y_1)$ et $\mathbf{x}_2 = (x_2, y_2)$.

La tâche principale de centrage de la droite épipolaire associée à ${}^d \mathbf{x}_m$ contraint deux degrés de liberté. Il reste donc quatre degrés de liberté disponibles pour l'application d'une tâche secondaire dédiée au parcours de la ligne de vue. En utilisant le formalisme de la redondance, le centrage successif des deux extrémités du segment ${}^d \mathbf{X}_1$ et ${}^d \mathbf{X}_2$ peut être réalisé sans perturber la régulation de la tâche principale.

Pour centrer les points ${}^d \mathbf{X}_i$, où $i \in \{1, 2\}$ dans le plan image de la caméra mobile, une tâche secondaire \mathbf{e}_2 est construite. Les informations visuelles considérées sont les projections \mathbf{x}_i des points virtuels ${}^d \mathbf{X}_i$ dans le plan image de la caméra mobile et le centre de l'image $\mathbf{x}^* = (0, 0)$. L'erreur à réguler est alors $\mathbf{e}_2 = \mathbf{x}_i - \mathbf{x}^*$. La matrice d'interaction \mathbf{L}_2 associée à cette tâche est la matrice d'interaction classique d'un point 2D [Espiau 92] :

$$\mathbf{L}_2 = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{pmatrix} \quad (3.21)$$

La commande appliquée au système pour permettre le parcours de la ligne de vue s'écrit :

$$\dot{\mathbf{q}} = -\lambda_1 (\mathbf{L}_1 {}^c \mathbf{V}_g \mathbf{J}(q))^+ (\mathbf{s}_1^* - \mathbf{s}_1) + \mathbf{P}_1 (-\lambda_2 (\mathbf{L}_2 {}^c \mathbf{V}_g \mathbf{J}(q))^+ (\mathbf{x}^* - \mathbf{x}_i)) \quad (3.22)$$

$$\dot{\mathbf{q}} = \mathbf{J}_1^+ \mathbf{e}_1 + \mathbf{P}_1 \mathbf{J}_2^+ \mathbf{e}_2 \quad (3.23)$$

avec \mathbf{P}_1 le projecteur sur la tâche principale.

Remarque : La matrice d'interaction associée à un point 2D n'est valide que si la profondeur Z est positive. Dès lors que le point 3D passe derrière le plan image, la convergence n'est plus prouvée. Il faut donc s'assurer que le point virtuel que l'on tente de centrer est bien positionné à l'avant du plan image. Si c'est le cas, on peut directement centrer ce point, sinon, il faut créer des points intermédiaires. La figure (3.6) illustre un cas pathologique dans lequel le centrage du premier point entraîne un positionnement du second point derrière le plan image.

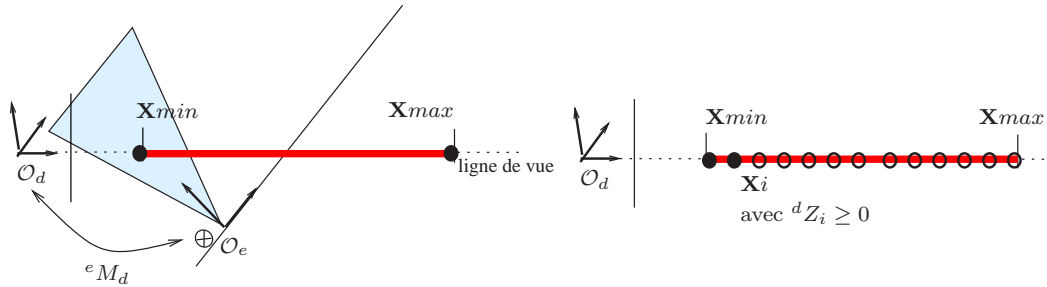


FIG. 3.6 – Selon la position relative des deux caméras, lorsque la tâche de centrage d'une extrémité du segment est réglée, la seconde extrémité peut être positionnée derrière le plan image de la caméra embarquée. La matrice d'interaction associée au second point n'est pas valide. La solution est d'effectuer n tâches de centrage successives pour garantir que les points 3D virtuels soient toujours devant le plan image.

Pour solutionner le problème, le parcours de la ligne est effectué en réglant séquentiellement n tâches de centrage de n points répartis régulièrement sur la ligne de vue, plutôt que de centrer une extrémité puis l'autre. Dans un premier temps, une des extrémités, telle que ${}^e Z \geq 0$, est centrée, puis successivement tous les points intermédiaires jusqu'à l'autre extrémité.

Comme nous le verrons dans la partie résultats, en plus d'assurer que le point courant a une profondeur ${}^e Z \geq 0$, centrer des points tout au long du segment permet d'acquérir des vues plus régulièrement. En effet, le centrage d'une extrémité puis de l'autre produit une vitesse de déplacement rapide lorsque l'erreur est importante et une vitesse faible à convergence; un grand nombre de vues sont acquises à proximité des extrémités du segment en comparaison à celles acquises sur le reste du segment.

3.2.3 Tâche à dimension variable : évitement des butées

Nous verrons, dans la partie consacrée aux validations expérimentales, que les tâches de centrage et de parcours de la ligne de vue engendrent principalement des mouvements de rotation. Les mouvements nécessaires à la réalisation des tâches \mathbf{e}_1 peuvent conduire les articulations du système en butée et perturber leur régulation. Si suffisamment de degrés de liberté sont disponibles, il est possible d'ajouter aux deux premières tâches détaillées, une troisième tâche d'évitement de butées.

L'approche la plus courante pour l'évitement de butées est la Méthode du Gradient Projeté [Liegeois 77, Samson 91] (voir figure 3.7). La tâche d'évitement de butée est définie par le gradient d'une fonction de coût qui atteint son maximum à l'approche des butées et sa dérivée doit s'annuler lorsque la fonction atteint son minimum. La fonction de coût peut par exemple avoir la forme illustrée par la figure 3.7. Cette méthode nécessite de choisir avec soin le rapport entre les gains de la tâche principale et celui de l'évitement de butées. En effet, il a été montré que si le gain de l'évitement de butée est trop faible la tâche d'évitement est inefficace. À l'inverse si ce gain est trop important, des oscillations apparaissent sur la position du robot.

Pour éviter ce problème, une solution est de ralentir ou stopper le mouvement des axes que la commande rapproche de leurs butées [Marchand 96]. Les autres degrés de liberté seront utilisés pour réguler les tâches de priorité supérieure. Cela revient à considérer le système comme ayant autant de

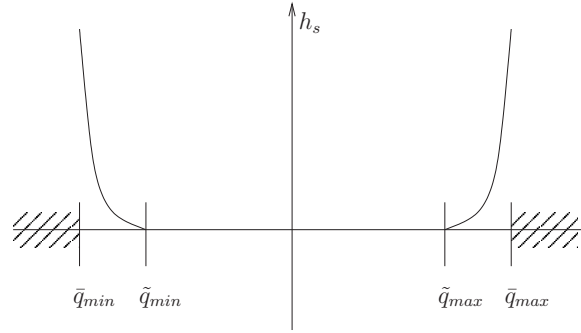


FIG. 3.7 – évolution de la fonction de coût en fonction de la position articulaire.

degrés de liberté en moins que d'axes en difficulté. Le rang de la tâche d'évitement de butée varie selon le nombre d'axes en phase critique.

L'algorithme d'évitement de butées implémenté est celui proposé dans [Chaumette 00]. Il est présenté plus en détail dans l'encadré (3.2.3).

Algorithme d'évitement de butée

- **Mesure de la position articulaire courante \mathbf{q}**
- **Calcul du jacobien courant** en fonction de la position articulaire courante \mathbf{q}
- **Calcul de la vitesse articulaire $\dot{\mathbf{q}}$** qui résulterait de l'application de la loi de commande
- **Prédiction des positions qu'atteindraient les axes** en un petit intervalle de temps δt si la vitesse $\dot{\mathbf{q}}$ leur était appliquée

$$\mathbf{q}_{pre} = \mathbf{q} - \lambda \dot{\mathbf{q}} \delta t$$

- **Évaluation des axes en difficulté.** Ce sont les axes dont la position q_{pre} est en dehors des limites articulaires tolérées et qui se rapprochent de leur butées. Compter le nombre K d'axes en difficulté.
- **Construction des matrices \mathbf{E} et \mathbf{S} .** \mathbf{E} est de dimension $(K \times 6)$ est construite en concaténant les lignes du jacobien de la tâche principale correspondantes aux axes en difficulté. La matrice \mathbf{S} est l'empilement des coordonnées du vecteur vitesse correspondant aux axes en difficulté.
- **Calcul de \mathbf{a}^* .** $\mathbf{a}^* = \mathbf{E}^+ \mathbf{S}$
- La vitesse à appliquer au système est

$$\dot{\mathbf{q}}' = \dot{\mathbf{q}} + \sum_{i=1}^{n_a} \mathbf{a}^* \mathbf{E}_{\bullet i}$$

FIG. 3.8 – Algorithme d'évitement de butée.

La commande peut être réécrite à partir de l'équation (3.22) et de l'équation (3.15) :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+ \mathbf{e}_1 + \mathbf{P}_1 \mathbf{J}_1^+ \mathbf{e}_1 + \sum_{i=1}^{n_a} \mathbf{a}_i \mathbf{E}_{\bullet i} \quad (3.24)$$

où n_a est la dimension du noyau de $\mathbf{J}_2^{\mathbf{A}} = (\mathbf{J}_1, \mathbf{J}_2)^{\top}$ et $\sum_{i=1}^{n_a} \mathbf{a}_i \mathbf{E}_{\bullet i}$ définit les mouvements qui per-

mettent d'éviter que les articulations en position critique ne rentrent en butée. \mathbf{E} est une base de $\mathbf{Ker J}_2^A$ et \mathbf{a} est un vecteur de gain qui annule le déplacement des axes en difficulté. \mathbf{a} est calculé automatiquement en résolvant le système :

$$\mathbf{F}\mathbf{a} = \mathbf{b} \text{ avec } \mathbf{F} = \begin{pmatrix} \vdots \\ \mathbf{E}_{k\bullet} \\ \vdots \end{pmatrix} \text{ et } \mathbf{b} = \begin{pmatrix} \vdots \\ (\mathbf{J}_2^A)_k \\ \vdots \end{pmatrix} \quad (3.25)$$

d'où $\mathbf{a}^* = \mathbf{F}^+\mathbf{b}$ et la loi de commande s'écrit :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+\mathbf{e}_1 + \mathbf{P}_1\mathbf{J}_2^+\mathbf{e}_2 + \sum_{i=1}^{n_a} \mathbf{a}^*\mathbf{E}_{\bullet i} \quad (3.26)$$

Malheureusement, il peut arriver que la loi de commande résultant de l'évitement des butées pour les axes k en difficulté entraîne certains des autres axes en butée. Pour éviter cette situation, il suffit d'itérer le processus. La loi de commande s'écrit finalement :

$$\dot{\mathbf{q}} = \mathbf{J}_1^+\mathbf{e}_1 + \mathbf{P}_1\mathbf{J}_2^+\mathbf{e}_2 + \sum_{i=1}^{n_a} \left(\mathbf{a}_i^* + \sum_{j=1}^{n_b} b_j \mathbf{F}_{\bullet j}^\perp \right) \mathbf{E}_{\bullet i} \quad (3.27)$$

Comme dans le cas précédent, le vecteur \mathbf{b} peut être déterminé en résolvant un système linéaire :

$$\sum_{j=1}^{n_b} \mathbf{b}_j \sum_{i=1}^{n_a} \mathbf{F}_{ij}^\perp \mathbf{E}_{ki} = - \left(\mathbf{J}_1^+\mathbf{e}_1 + \mathbf{P}_1\mathbf{J}_2^+\mathbf{e}_2 + \sum_{i=1}^{n_a} \mathbf{a}_i^* \mathbf{E}_{\bullet i} \right) \quad (3.28)$$

L'utilisation de la forme itérative permet de trouver une solution au problème d'évitement de butées, si une solution existe. Cette méthode permet de s'abstraire du calcul du gain qui est critique lors de l'utilisation de la méthode du gradient projeté. Mais elle présente un inconvénient : elle provoque des discontinuités dans la commande aux moments des changements de rang, c'est-à-dire à chaque fois qu'un axe entre ou sort de la zone critique. Pour atténuer les discontinuités engendrées par ces changements de rang, une solution est d'atténuer la vitesse appliquée à un axe dès qu'il approche de la zone critique [Marchand 96]. Notre loi de commande ne repose pas sur le suivi d'indices visuels dans la vue embarquée, le seul suivi peut éventuellement être fait dans la vue déportée dans le cas d'un objet mobile. Les discontinuités que l'évitement de butées fait apparaître ne sont donc pas gênantes pour le traitement de l'image, tant que l'odométrie du système n'est pas perturbée. En revanche, ces à-coups peuvent être dérangeants dans le cadre de notre application et effrayer la personne qui utilise l'outil. Il faut donc veiller, à terme, à lisser ces discontinuités et à proposer un évitement de butées qui n'engendre pas de mouvements brusques du bras.

3.3 Résultats expérimentaux

La commande proposée dans le paragraphe précédent a été testée sur les deux plateformes robotiques présentées sur la figure (3.9) : le robot cartésien (AFMA Robotics) de l'équipe Lagadic à l'INRIA Rennes et sur le robot anthropomorphe (Staubli RX90) dans l'équipe LTC au CEA-LIST. Le robot AFMA a six degrés de liberté : les trois premiers correspondent aux trois translations et les trois autres aux trois rotations. Le robot Staubli RX90 est un bras anthropomorphe dont les six articulations sont rotoïdes. Ce paragraphe présente les résultats obtenus :

- dans un cas où l'évitement de butée n'est pas nécessaire ;
- dans un cas où l'évitement de butée est nécessaire ;
- dans le cas où l'objet recherché est en mouvement.

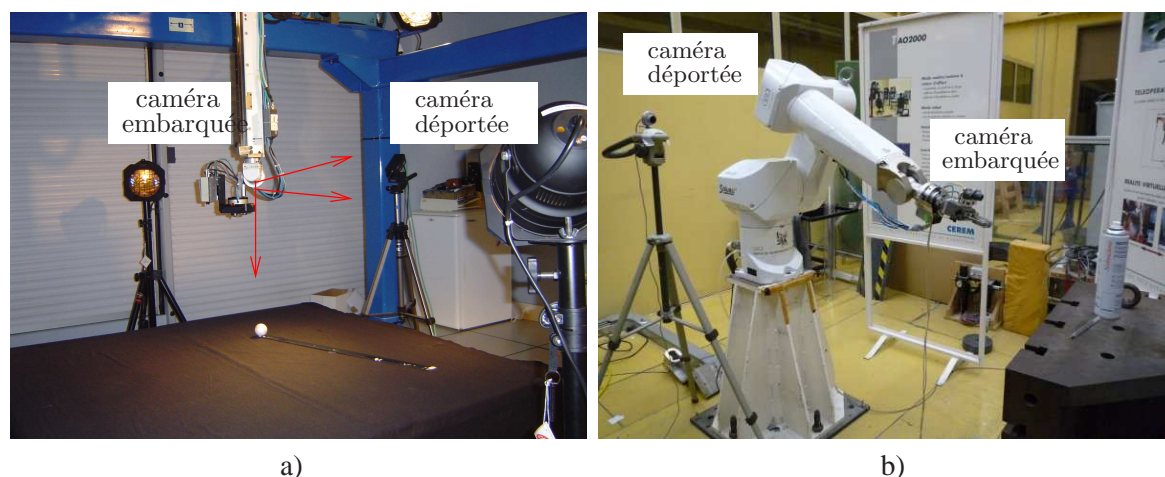


FIG. 3.9 – À gauche : le robot cartésien de l'équipe Lagadic à l'INRIA Rennes. À droite le robot 6 axes Staubli RX90, du LTC au CEA-LIST

3.3.1 Parcours de la ligne de vue

Dans le cas le plus simple, l'application de la loi de commande permet le parcours de la ligne de vue en trois étapes :

1. centrer la ligne de vue
2. centrer l'extrémité du segment qui est en avant du plan image de la caméra embarquée
3. centrer l'autre extrémité

Cette méthode a été appliquée sur le robot cartésien AFMA Robotics de l'INRIA Rennes. Elle est particulièrement bien adaptée à ce type de bras manipulateur puisque la caméra surplombe la scène. Les deux extrémités du segment sont donc, la plupart du temps, à l'avant du plan image de la caméra embarquée. D'autre part, ce bras n'entre que rarement en butée. Généralement, seuls les derniers degrés de rotation posent problème, principalement à cause de la caméra que l'effecteur porte et de son câblage.

La figure 3.10 présente un résultat typique obtenu en appliquant une loi de commande à deux tâches, une tâche de centrage de la ligne épipolaire et une tâche de centrage des extrémités du segment d'intérêt. Dans cet exemple, la scène contient un masque en bois peint posé sur une nappe à motif répétitif. L'utilisateur sélectionne un point du masque. Dans une première phase, seule la tâche prioritaire de centrage de la ligne de vue s'exprime (caméras bleues et les trois premières images). La ligne de vue courante (en vert) s'aligne petit à petit sur la consigne (en rouge). Lorsque l'erreur de la tâche primaire devient suffisamment faible, la tâche secondaire permettant le parcours de la ligne de vue s'exprime à son tour.

La figure 3.11 présente l'évolution des erreurs associées à ces deux tâches et des vitesses cartésiennes. Les vitesses sont exprimées dans le repère de la caméra embarquée. La tâche secondaire s'exprime dès que la tâche primaire est suffisamment régulée. Cette tâche secondaire induit principalement des mouvements de rotation autour d'un axe normal au plan épipolaire, qui est l'axe j du repère de la caméra embarquée. Pour limiter les déplacements du bras à proximité d'une personne, les mouvements induits par cette seconde tâche peuvent être limités à des mouvements de rotation.

Centrer successivement les deux extrémités du segment d'intérêt permet de parcourir la ligne de vue. Cependant cette méthode présente deux inconvénients. Le premier est dû à la décroissance exponentielle de l'erreur et à l'acquisition des images à intervalles réguliers : loin de la convergence, la caméra se déplace rapidement, deux vues successives sont donc très espacées, alors qu'un nombre d'images plus important est acquis à proximité de la convergence, lorsque la vitesse de la caméra

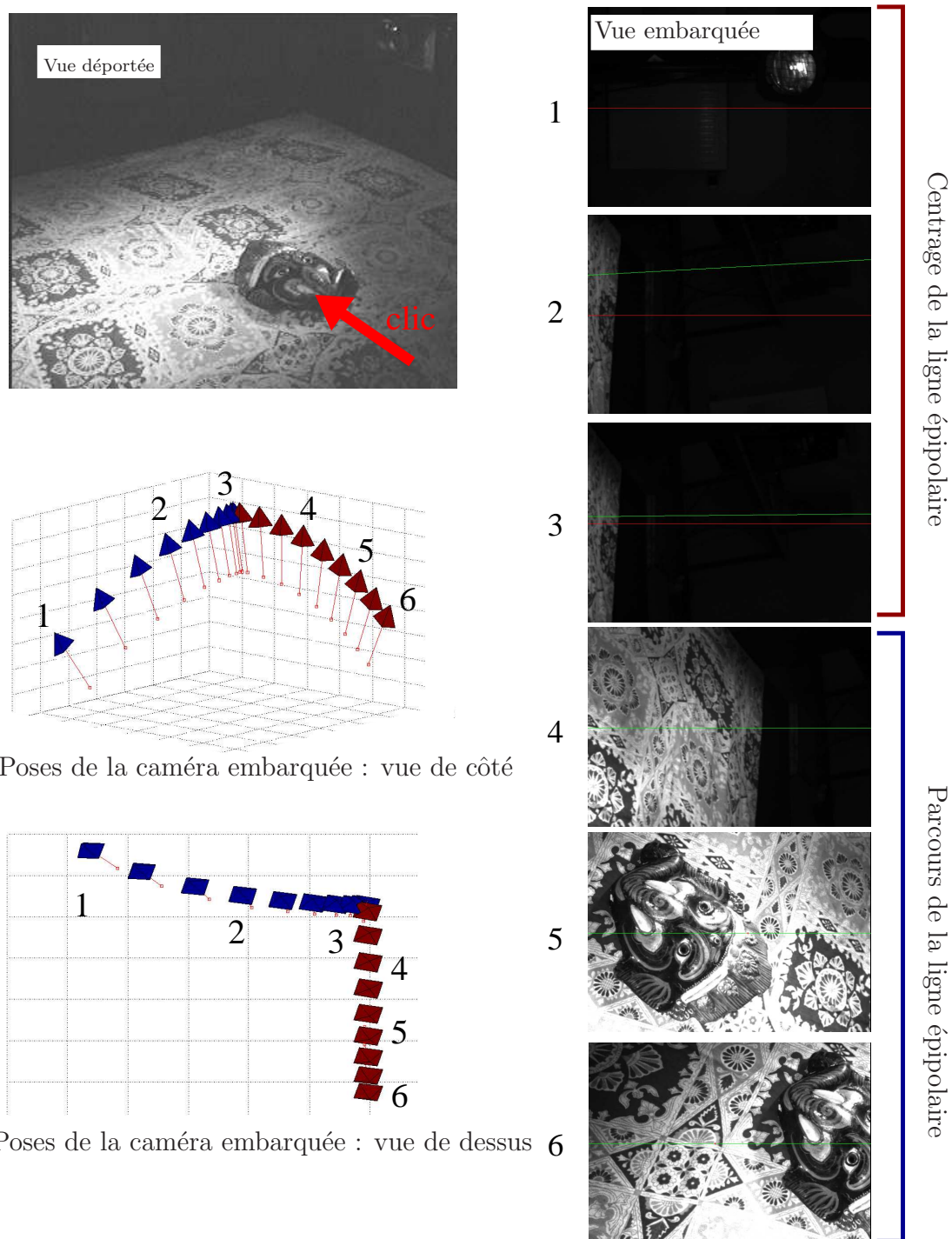


FIG. 3.10 – Asservissement visuel sur la ligne épipolaire : En haut à gauche, la vue de la caméra déportée. Un point du masque est sélectionné par l'utilisateur. Les deux autres images du haut représentent les différentes positions occupées par la caméra au cours du parcours de la ligne de vue. Les poses représentées en bleu correspondent à la phase de centrage de la ligne de vue. Les poses acquises en rouge, correspondent au parcours de la ligne de vue à proprement parler. En bas, les trois premières images sont acquises au cours du centrage et les trois images suivantes, au long du parcours de la ligne de vue. La ligne épipolaire associée au clic apparaît en vert, la ligne rouge est la consigne de centrage.

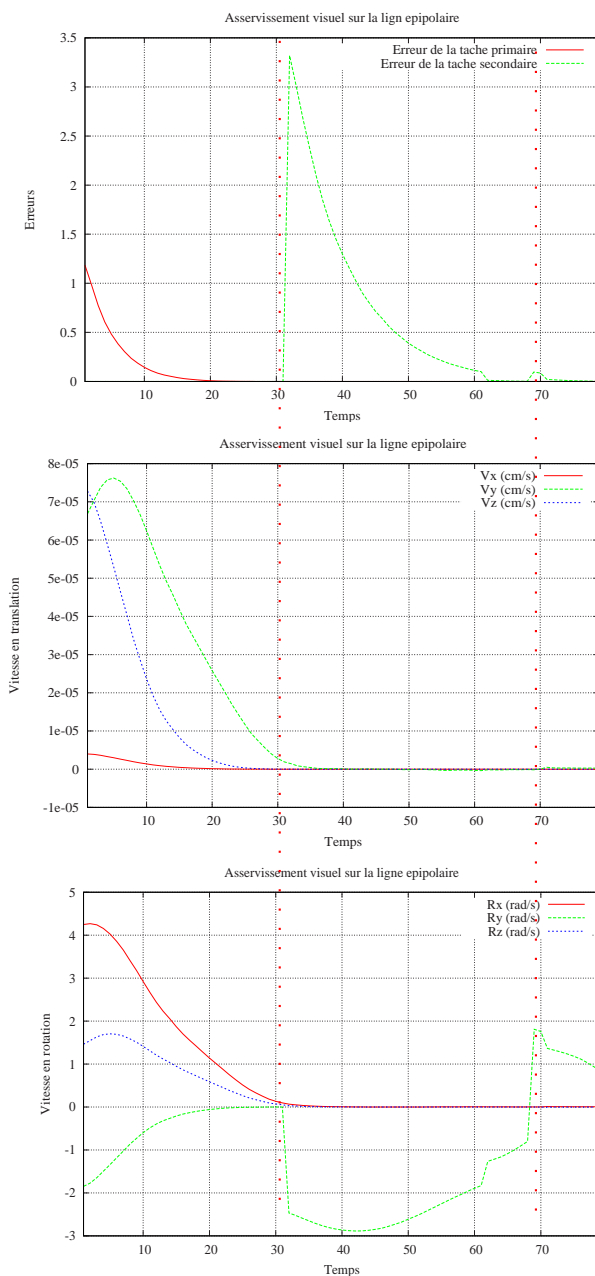


FIG. 3.11 – Asservissement visuel sur la ligne épipolaire : en haut, les erreurs des tâches primaire (rouge) et secondaire (vert) ; au milieu, les vitesses de translation ; en bas, les vitesses de rotation. On peut noter que la tâche secondaire engendre principalement un mouvement de rotation Ω_x autour de l'axe \mathbf{j} . En effet, lorsque la tâche primaire est réglée, l'axe \mathbf{j} est normal au plan épipolaire. Une rotation autour de cette axe est donc suffisante pour parcourir la ligne de vue passant par l'objet.

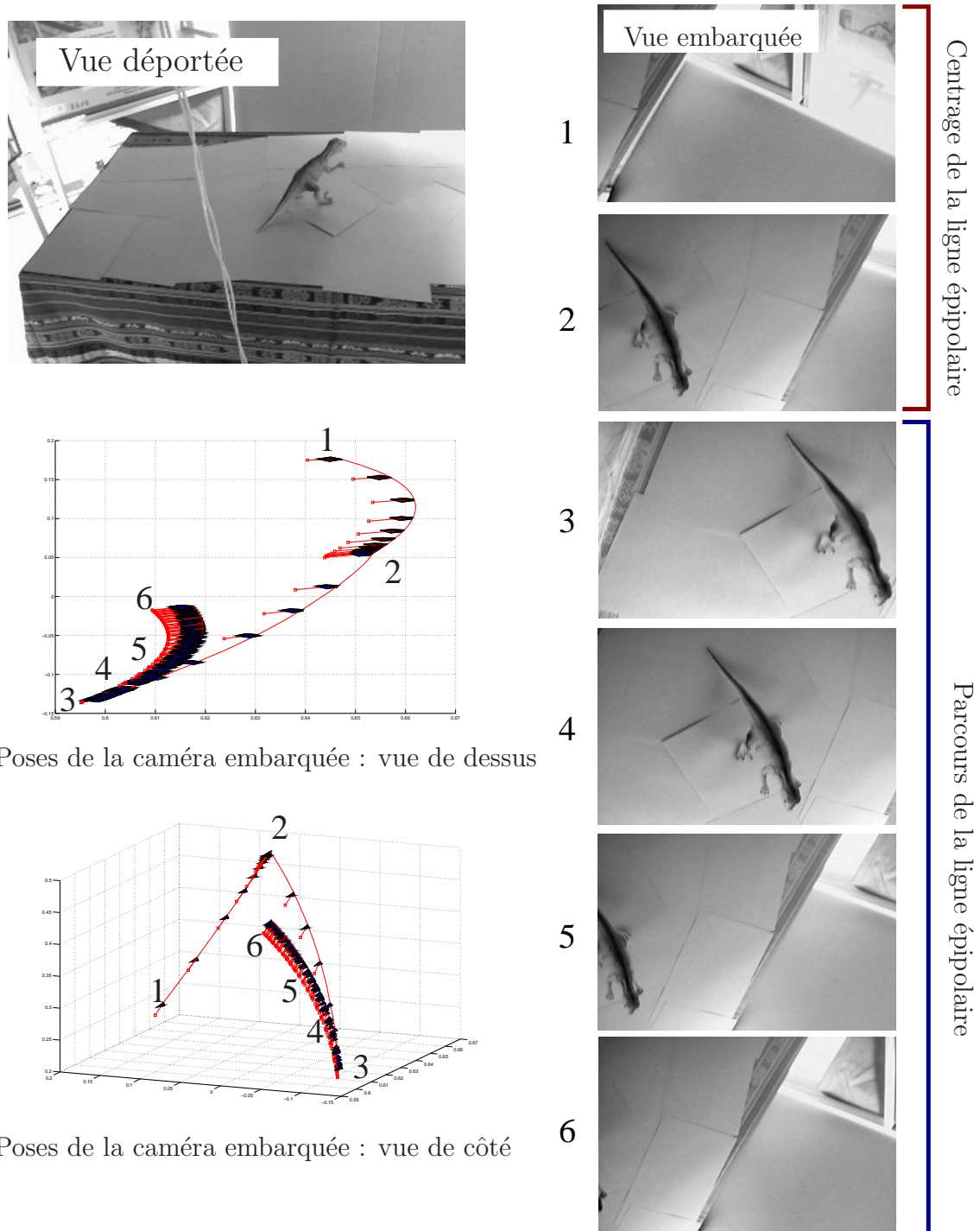


FIG. 3.12 – Asservissement visuel sur la ligne épipolaire : En haut à gauche, la vue de la caméra déportée. Un point du dinosaure est sélectionné par l'utilisateur. Les deux autres images du haut représentent les différentes positions occupées par la caméra au cours du parcours de la ligne de vue. Les poses représentées en bleu correspondent à la phase de centrage de la ligne de vue. Les poses acquises en rouge, correspondent au parcours de la ligne de vue à proprement parler. En bas, les trois premières images sont acquises au cours du centrage et les trois images suivantes, au long du parcours de la ligne de vue. La ligne épipolaire associée au clic apparaît en vert, la ligne rouge est la consigne de centrage.

embarquée est faible. La solution à ce problème peut être trouvée en jouant sur le déplacement de la caméra ou sur la fréquence d'acquisition des images.

Par exemple, une première solution est d'utiliser un gain adaptatif qui permet d'accélérer la convergence en augmentant le gain lorsque l'erreur est devenue faible. Ainsi, un nombre d'images moins important que précédemment est acquis aux extrémités du segment. Cette solution est cependant une solution partielle puisqu'il y aura toujours un plus grand nombre d'images des extrémités du segment que de son milieu.

Une seconde solution peut être trouvée au niveau de la commande et consiste à utiliser une commande à mesure glissante : plutôt que de fixer x successivement aux extrémités du segment, $x = x(t)$ peut varier d'une extrémité à l'autre au cours du temps. Ainsi, la caméra parcourt la ligne de vue et acquiert des images à intervalle spatial régulier.

Une autre solution peut être trouvée dans les techniques de traitement de l'image. Une image de la séquence n'est traitée que si elle présente suffisamment de différence avec l'image précédente. Ainsi, les images qui se superposent au voisinage de la convergence ne sont pas prises en compte. L'inconvénient de cette méthode est qu'elle nécessite l'extraction et la comparaison d'informations dans toutes les images de la séquence, ce qui ajoute un temps de traitement supplémentaire.

Le deuxième inconvénient de cette méthode est que le centrage de la première extrémité peut faire passer la seconde extrémité du segment derrière le plan image de la caméra déportée. Si bien que la tâche de centrage de la seconde extrémité n'est pas réalisable ($Z < 0$) et le parcours de la ligne échoue. Pour éviter ce problème, la commande à mesure glissante est une solution. Dans un premier temps, l'extrémité qui se trouve à l'avant du plan image de la caméra embarquée est centrée. Ensuite, cette méthode permet de garantir que le point x courant est à l'avant du plan image de la caméra puisqu'il part de l'extrémité centrée courante qui est à l'avant du plan image.

Dans la suite de ce chapitre, une autre solution est proposée pour résoudre le problème du recouvrement des vues et de la redondance d'informations qu'il engendre.

3.3.2 Parcours de la ligne de vue en évitant les butées

Lorsque le parcours de la ligne de vue est effectué par bras anthropomorphe, par exemple le Staubli RX90 du CEA, ses butées doivent être prises en compte pour réaliser la tâche. Sinon, dès qu'un axe atteint une position proche de ses butées, le système n'autorise plus de mouvement sur ces articulations et l'exécution de la tâche est interrompue. Pour tenir compte des butées, trois tâches sont effectuées :

1. Centrage de la ligne de vue ;
2. Parcours la ligne de vue ;
3. Évitement des butées.

La figure 3.13 présente l'évolution de la commande articulaire dans un cas où le robot est en butée et la figure 3.14, présente les résultats obtenus pour la même situation en ajoutant une troisième tâche d'évitement de butées. L'évitement de butée permet de réaliser le parcours de la ligne de vue. On peut cependant remarquer l'apparition de discontinuités dans la commande qui peuvent provoquer des déplacements brutaux du bras. Ces discontinuités ne perturbent pas la tâche principale. Le parcours de la ligne de vue est réalisé et l'objet est vu dans quelques images. Tant que ces discontinuités ne perturbent pas l'odométrie du système, la méthode de localisation d'un objet inconnu que nous proposons fonctionne. Cependant, pour que cette méthode puisse être portée sur une assistance technique d'aide à la saisie, ces déplacements brutaux devront être lissés pour des raisons de sécurité et de confort.

3.3.3 Suivi d'un objet en mouvement

Le loi de commande que nous avons présentée est robuste à des mouvements relatifs de l'objet et de la caméra déportée. Tant que l'objet est dans l'image de la caméra déportée, il peut être suivi et le parcours de la ligne de vue est réalisé (voir les figures 3.16 et 3.17).

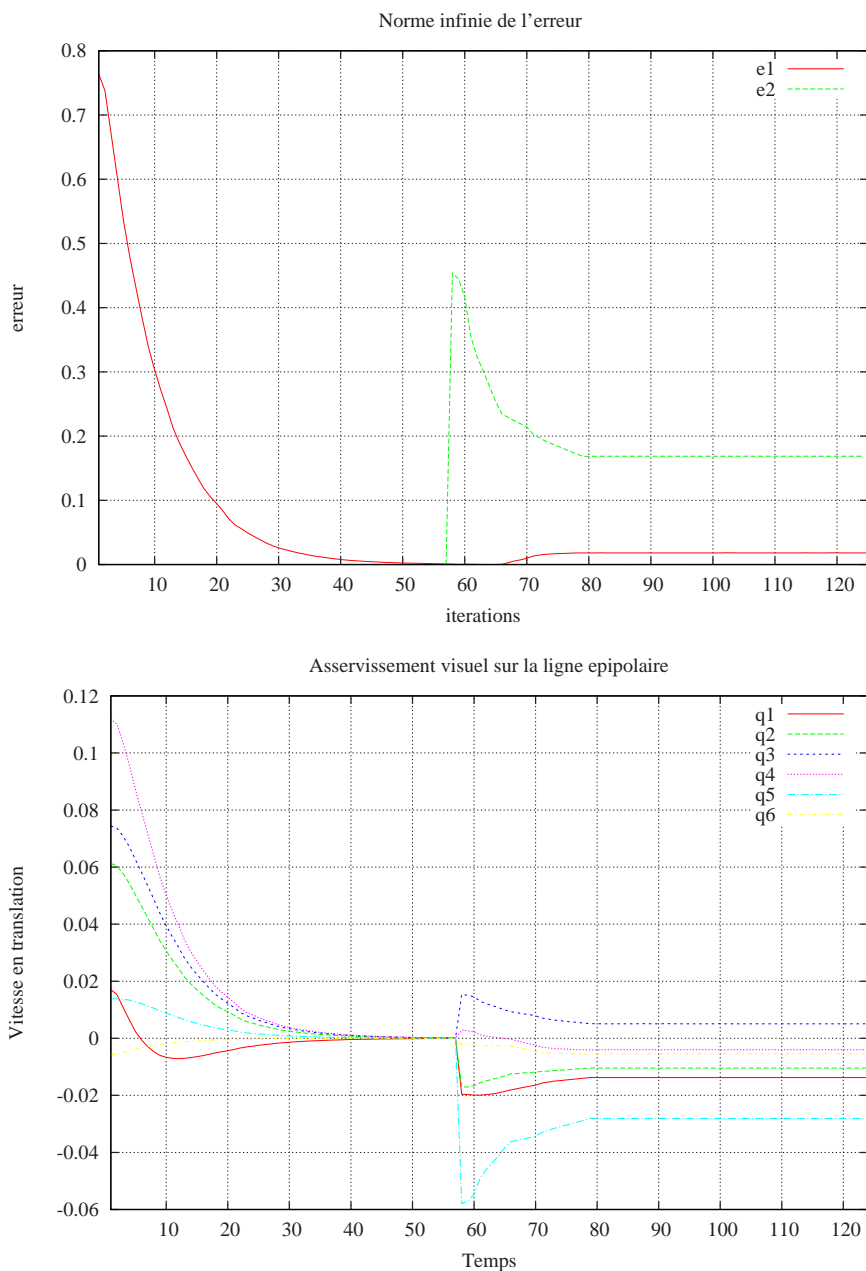


FIG. 3.13 – Le bras entre en butée au bout de 58 itérations. La tâche cesse alors de converger. La commande en vitesse reste constante. En réalité ces vitesses ne sont pas appliquées au système qui est en butée. La convergence est interrompue.

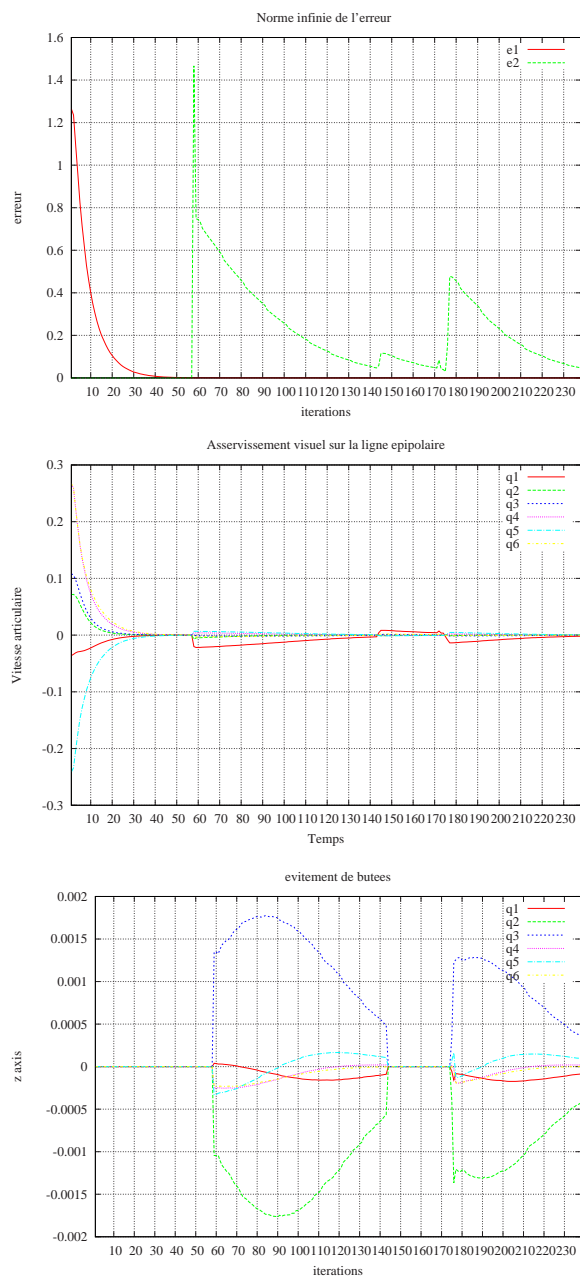


FIG. 3.14 – Asservissement visuel sur la ligne épipolaire : en haut, les erreurs des tâches primaires (rouge) et secondaire (vert) ; en bas, les vitesses articulaires. Les discontinuités dans la commande sont due au centrage de i points successifs tout au long du segment. Elles peuvent être lissées en faisant varier la position du point courant continuellement plutôt qu'utiliser un ensemble de points discrets.

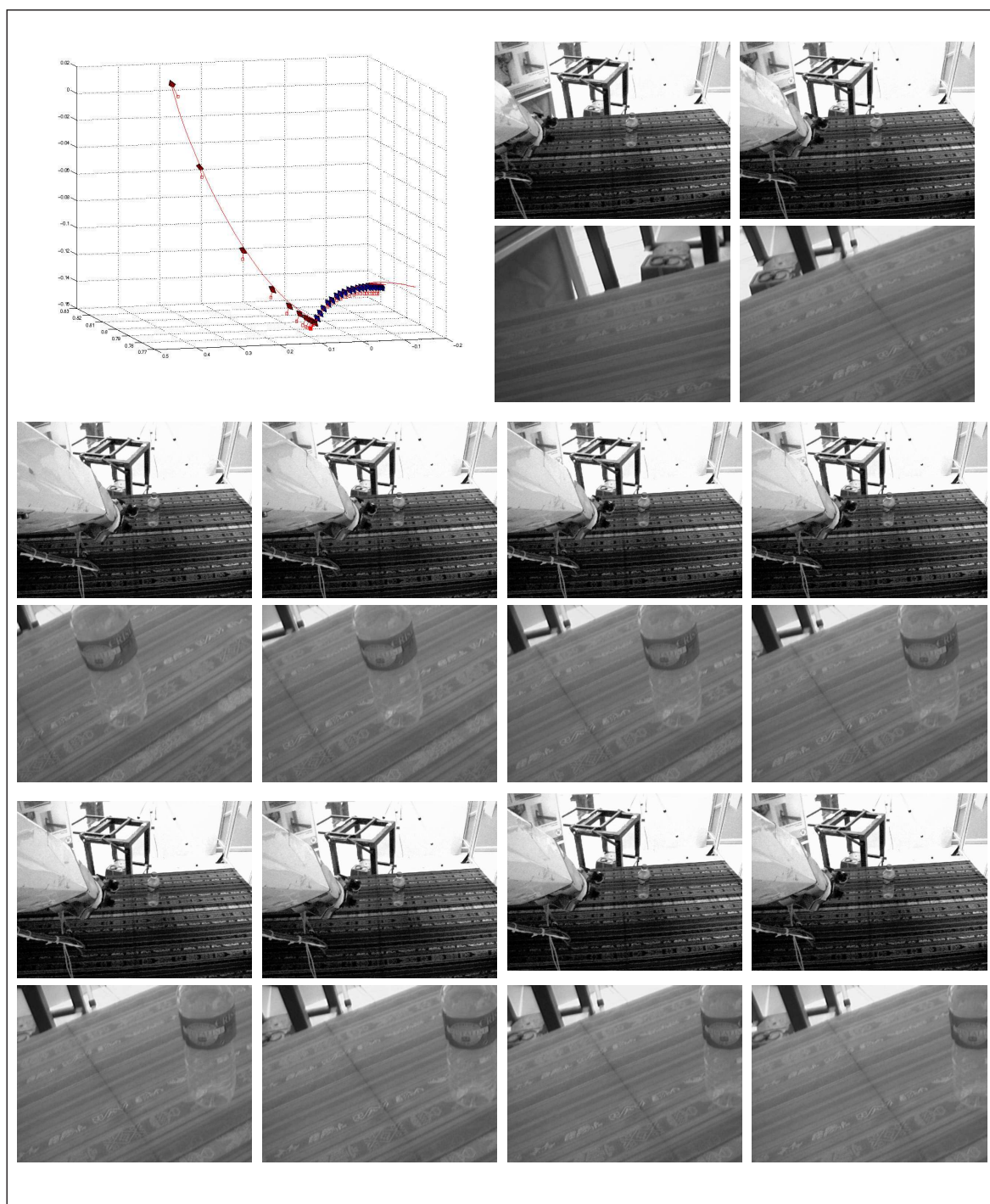


FIG. 3.15 – Images obtenue lors de l'envoi de la commande décrite par la figure 3.14. En rouge les caméra avant l'évitement de butées et en bleu les poses après l'évitement de butée

Lorsque l'objet est en mouvement par rapport au repère de la caméra déportée, des erreurs de traînage dues au suivi de l'objet peuvent apparaître. La décroissance des erreurs des tâches est affectée par ces erreurs de traînage. Ces effets n'apparaissent pas sur la figure de résultat car l'objet se déplace quasiment sur la ligne de vue. Son mouvement ne perturbe donc que très peu le centrage et le parcours de la ligne épipolaire. Cependant le suivi de l'objet peut échouer dès lors qu'une occultation se produit, typiquement si le bras passe entre la caméra déportée et l'objet (voir figure 3.18). Sans traitement des occultations, la méthode de suivi peut échouer.

Conclusion

Le *clic* donne une information lacunaire sur la localisation de l'objet : il permet de calculer les paramètres d'une droite 3D qui passe par le centre de la caméra déportée et le point sélectionné. La distance qui sépare un objet inconnu de la caméra déportée ne peut être calculée à partir de la seule information donnée par la *clic*. Une seconde vue de l'objet est nécessaire pour obtenir cette information. Cependant, comme aucune information de profondeur n'est disponible, il faut acquérir plusieurs images permettant de visualiser des portions de la ligne de vue.

La localisation de l'objet se limite à un segment 3D défini par l'intersection entre la ligne de vue correspondant au *clic* et l'espace de travail du bras. La loi de commande présentée dans ce chapitre permet de déplacer la caméra embarquée de façon à ce qu'elle acquiert des images pour visualiser la totalité du segment 3D qui porte l'objet d'intérêt. L'utilisation d'une boucle fermée rend la méthode robuste à des mouvements relatifs de l'objet et de la caméra déportée. Cela permet de relâcher l'hypothèse d'un système fixe. Ainsi, si le bras et les deux caméras sont montés sur un fauteuil roulant, et que la personne déplace le fauteuil en cours de saisie, il est possible de suivre l'objet dans la vue de la caméra déportée et de mettre à jour la position de la ligne de vue en utilisant les coordonnées courantes du point sélectionné.

Finalement, dès que la ligne de vue est centrée dans l'image, le mouvement permettant le parcours de la ligne de vue est principalement un mouvement de rotation autour d'un axe normal au plan épipolaire. En d'autres termes, lorsque la ligne de vue est centrée dans la vue de la caméra embarquée, cela revient à effectuer une rotation autour de l'axe \mathbf{j} de la caméra embarquée. Limiter les mouvements de la caméra embarquée à des rotations présente deux avantages :

- la caméra n'occulte pas la vue de la caméra déportée, le suivi de l'objet est alors possible ;
- la caméra se déplace peu, les mouvements du bras sont discrets et n'effraient pas l'utilisateur ;

Le chapitre suivant présente la méthode que nous avons développée pour localiser l'objet d'intérêt sur la ligne de vue à partir des informations extraites des images acquises au fur et à mesure du parcours de la ligne de vue.

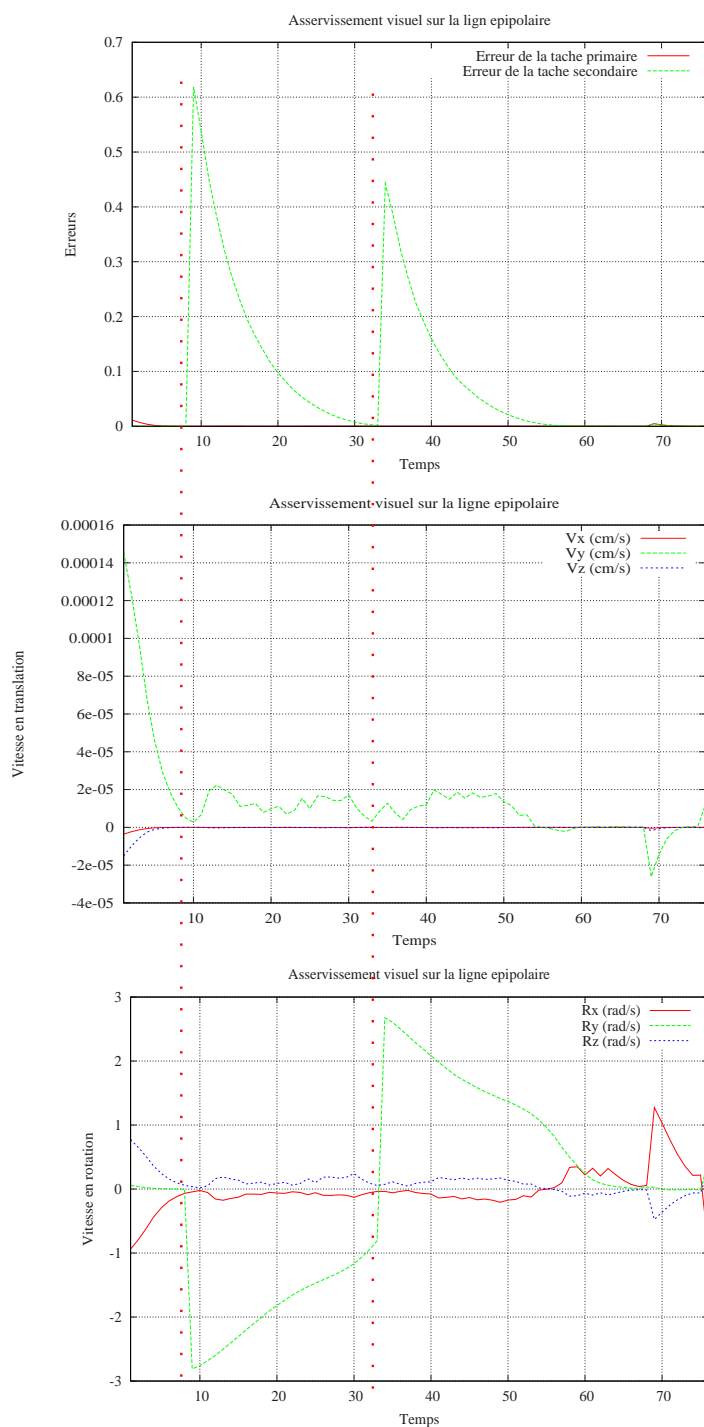


FIG. 3.16 – Suivi d'un objet en mouvement. Évolution des erreurs et des vitesses cartésiennes. Le déplacement de l'objet perturbe la décroissance exponentielle des erreurs. Les tâches sont réglées à une erreur de traînage près.

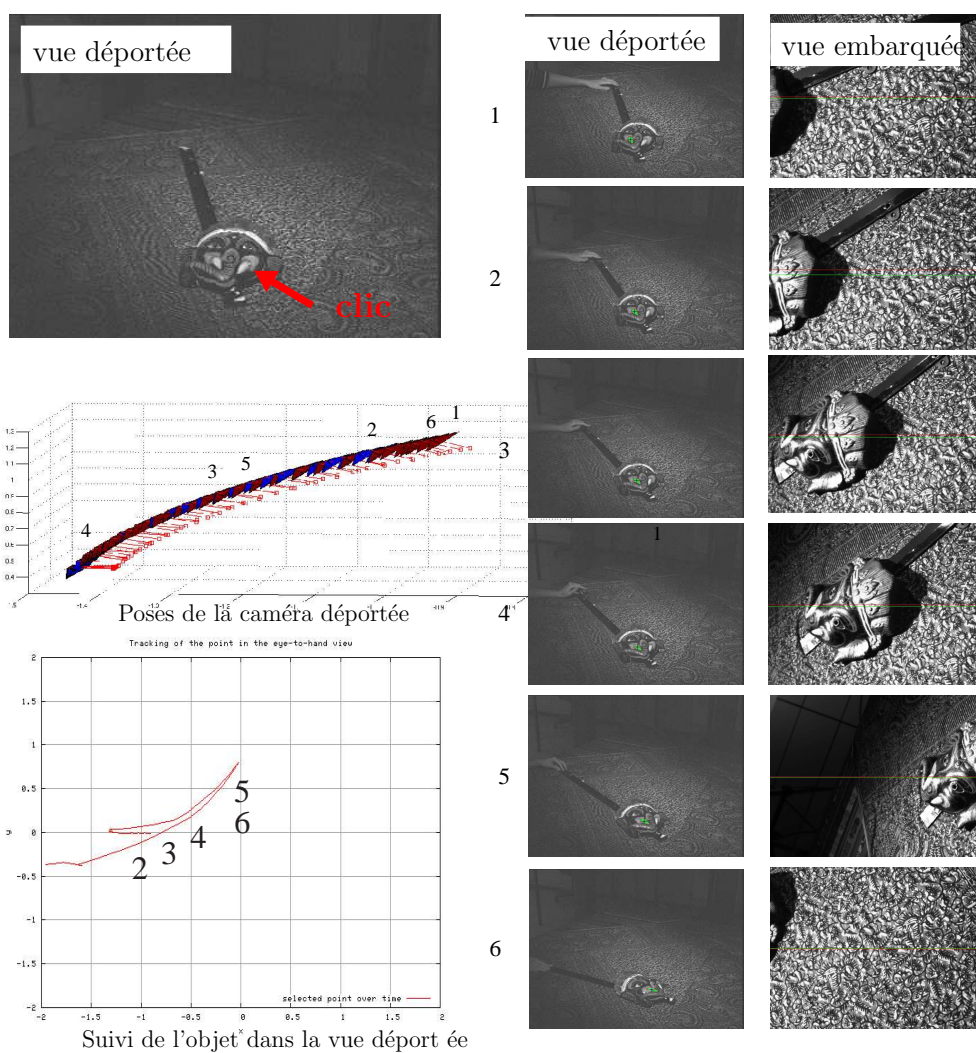


FIG. 3.17 – Suivi d'un objet en mouvement. En haut à gauche l'image déportée initiale où l'objet est sélectionné. En haut à droite le parcours de la caméra embarquée. La deuxième ligne contient les vues successives de la caméra déportée. La troisième ligne le motif suivi dans ces vues. Enfin la dernière ligne présente les vues acquises depuis la caméra embarquée.

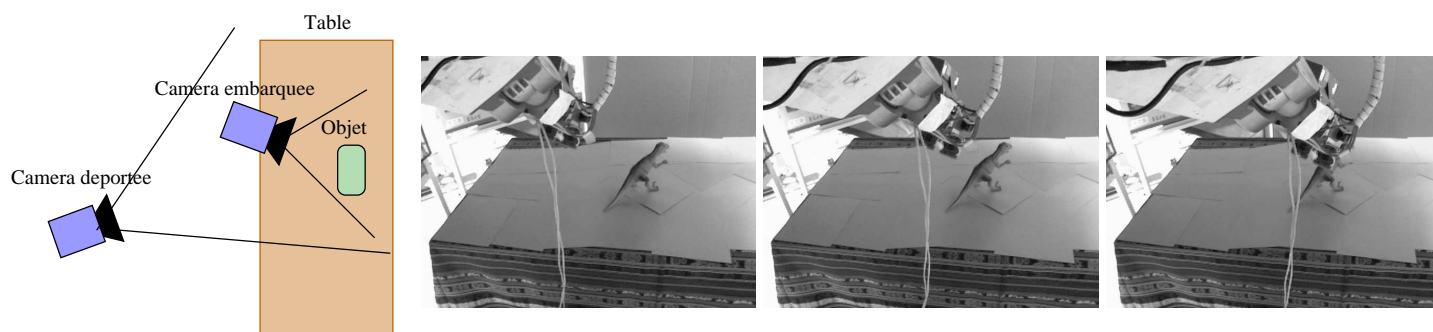


FIG. 3.18 – Si la caméra embarquée passe entre la caméra déportée et l'objet, alors le motif avoisinant le clic ne peut plus être suivi.

Localisation de l'objet sur une ligne de vue

Dans le chapitre précédent, nous avons présenté une commande basée sur la géométrie épipolaire permettant de déplacer la caméra embarquée pour acquérir des images le long de la ligne de vue associée à un *clic* dans l'image déportée (voir la figure 4.1)). Ce chapitre présente une méthode permettant de détecter l'objet dans ces vues. Connaissant la position de la caméra lors de l'acquisition de chacune des images ainsi que ses paramètres intrinsèques, nous montrerons que l'objet peut être grossièrement localisé dans l'espace .

4.1 Un problème de reconnaissance d'apparence

À l'instant initial, la caméra déportée voit l'objet et la caméra embarquée est dans une position quelconque. Rien ne garantit *a priori* que l'objet se projette dans la première image acquise par la caméra embarquée, par contre, il est situé à proximité de la ligne de vue associée au *clic*, il apparaît donc dans plusieurs images successives acquises lors du parcours de la ligne de vue (voir le chapitre précédent).

Les informations les plus pertinentes se trouvent au voisinage du *clic*. Connaissant la position du *clic* dans la vue de la caméra déportée, une vue de l'objet est donc disponible à son voisinage. Ainsi, repérer l'objet sur la ligne de vue revient à résoudre un problème de reconnaissance d'objet en recherchant, dans les vues de la caméra embarquée, des structures ou des éléments identiques à ceux qui se trouvent à proximité du *clic* dans la vue de la caméra déportée.

Tandis que la caméra déportée donne une vue globale de la scène et est relativement éloignée de l'objet, la caméra embarquée peut être déplacée à l'intérieur de la scène et en donne une vue de détails. L'objet est donc vu depuis deux points de vue radicalement différents. La méthode de reconnaissance doit donc être robuste à la fois aux changements d'échelle et aux changements d'orientation.

Si l'objet est relativement petit dans l'image de caméra déportée, comme c'est le cas sur la figure (4.1), il pourrait être intéressant de limiter l'extraction de données à la zone de l'image la plus pertinente, pour gagner en temps de calcul et en précision de localisation. Nous reviendrons sur les méthodes de segmentation d'un objet inconnu dans le chapitre 8. Dans ce chapitre, pour ne pas imposer de contrainte supplémentaire sur la scène et les objets, nous proposons une méthode de localisation

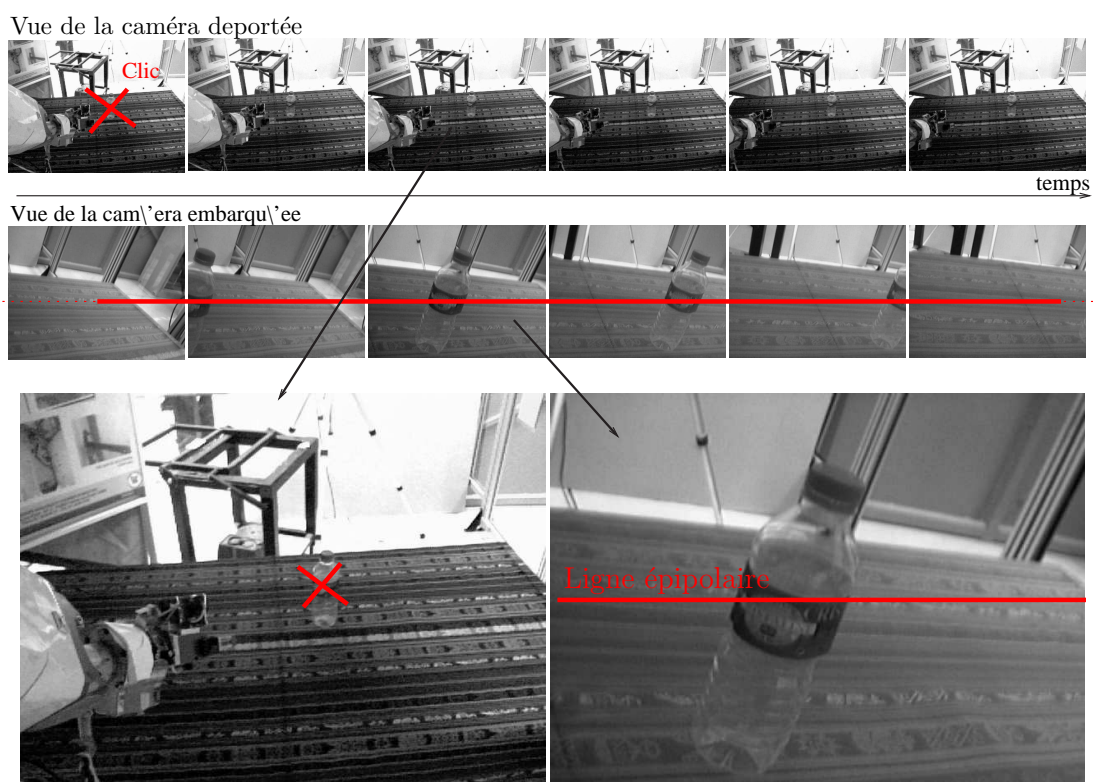


FIG. 4.1 – Images acquises par les deux caméras au cours du parcours de la ligne de vue associée au clic. En haut, les images acquises par la caméra déportée qui contiennent l'objet. En bas la vue de la caméra embarquée dans laquelle l'objet se projette par intermittence.

qui n'utilise pas de segmentation préalable de l'objet sur le fond. Les informations sont extraites dans toute l'image et inversement pondérées par leur distance au *clac* : plus elles sont proches du *clac*, plus elles sont considérées comme fiables, et vice et versa.

4.2 Méthode de reconnaissance robuste

Ce paragraphe présente une méthode de reconnaissance robuste aux changements d'échelles et d'orientation, reposant sur l'apparence de l'objet au voisinage du *clac*. Dans un premier temps, la méthode de calcul des descripteurs visuels est présentée, puis une méthode de mise en correspondance permettant de déterminer l'emplacement de l'objet dans les images et enfin l'information que cette mise en correspondance donne sur la profondeur de l'objet sur la ligne de vue. Par la suite, cette information sera convertie sous forme d'une densité de probabilité sur la position de l'objet sur la ligne de vue, nous ramenant ainsi à un problème unidimensionnel.

4.2.1 Invariants robustes aux changements d'échelle et d'orientation

L'objectif est de décrire le voisinage du *clac* par un ensemble de descripteurs qui pourront être reconnus, dans la vue embarquée, sous un changement d'orientation et d'échelle importants. Les descripteurs photométriques locaux (SIFT [Lowe 04], SURF, FERNS, GLOH) répondent très bien à ce type de problème : ils sont discriminants, robustes aux occultations et ne demandent pas de segmentation. Ils sont calculés directement à partir de l'information contenue dans des images numériques au voisinage de points d'intérêt : intensité des pixels, couleur, texture, contour, orientation des gradients, etc. Différents types de descripteurs ont été développés et leur efficacité dépend du choix de la méthode d'extraction des points d'intérêt, de la construction du descripteur et de la technique de mise en correspondance. Dans notre cas, nous nous intéresserons à des descripteurs basés sur les niveaux de gris.

Les principaux descripteurs utilisés dans le contexte de la reconnaissance ou la mise en correspondance d'une même scène ou d'un objet sous des orientations différentes ont été comparés par Mikolajczyk et Schmidt dans une étude parue en 2005 [Mikolajczyk 05]. Il apparaît que les descripteurs SIFT¹ (*Scale Invariant Features Transform*) [Lowe 04] et leur extension, les descripteurs GLOH (*Gradient Location and Orientation Histogram*) permettent d'obtenir la meilleure qualité de mise en correspondance.

Les SIFT sont largement utilisés en vision par ordinateur pour reconnaître des structures sous des changements de points de vue et d'échelle importants. Ils sont par ailleurs robustes au changement d'illumination et au bruit. D'autre part, ils sont très discriminants et permettent des mises en correspondance robustes avec un faible taux d'erreur. Enfin, la reconnaissance d'objet basée sur les SIFT est robuste aux occultations partielles [Mikolajczyk 05].

L'encadré (4.2) et la figure (4.3) présentent l'algorithme des SIFT qui permet d'extraire des points d'intérêt multi échelles et de construire les descripteurs associés sous la forme de vecteurs de 128 éléments qui en décrivent le voisinage. À chaque point d'intérêt sont attachées une position, une orientation et une échelle.

4.2.2 Mise en correspondance des descripteurs SIFT

Pour retrouver dans une image embarquée l'objet qui se trouve au voisinage du *clac* dans l'image déportée, il faut mettre en correspondance les points extraits dans chacune des vues, c'est-à-dire pouvoir comparer deux descripteurs et déterminer s'il s'agit du même point.

¹L'implémentation de l'algorithme utilisé dans le cadre de cette thèse est celle de Rob Hess : <http://web.engr.oregonstate.edu/~hess/index.html>

L'algorithme des Scale Invariant Features Transform [Lowe 04].

L'algorithme est divisé en quatre étapes :

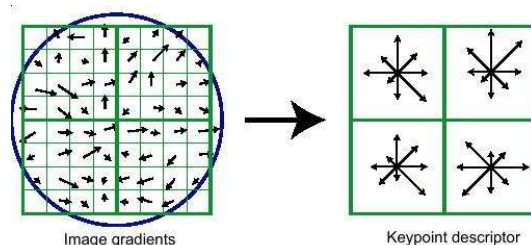
Détection des extremums dans l'espace d'échelle. Les points clés sont détectés en calculant les extremums de l'espace d'échelle construit à partir du calcul de différence de gaussiennes (DoG) sur les octaves d'une pyramide gaussienne comme illustré par la figure (4.3).

Localisation des points d'intérêt. Cette méthode de sélection produit trop de points d'intérêt dont certains ne sont pas des indices stables. L'étape de localisation repose sur une approximation de la fonction DoG au voisinage des extremums par un développement de Taylor qui permet d'affiner la localisation des points et de leur attribuer une échelle et un poids relatif à la courbure principale. Ainsi, les points trop proches l'un de l'autre, les points qui ont un trop faible contraste et ceux qui sont situés à un emplacement quelconque sur un contour sont éliminés.

Calcul de l'orientation. Une ou plusieurs orientations relatives à la direction des gradients et à leur amplitude sont associées à chaque point d'intérêt, ce qui assure l'invariance à l'orientation. L'amplitude et la direction du gradient sont calculées pour chaque pixel au voisinage du point d'intérêt et un histogramme de dimension 36 (soit $36 \times 10^\circ$) est construit. Chaque échantillon est pondéré par l'amplitude de son gradient et sa distance au point.

Les pics de l'histogramme correspondent aux orientations dominantes qui sont ajoutées au descripteur. Si plusieurs orientations sont concurrentes, alors un autre point clé, copie du premier est créée et la deuxième orientation lui est affectée.

Construction des descripteurs.



L'amplitude et la direction du gradient sont calculées pour les pixels voisins d'un point d'intérêt. L'amplitude est pondérée par une gaussienne centrée sur le point. Les résultats sont stockés dans quatre histogrammes d'orientation de dimension 8. Sur la figure ci dessus, la longueur des flèches représente la valeur de l'histogramme pour cette orientation. La figure présente un descripteur de 2×2 histogrammes de 8 dimensions. Les SIFT sont en fait implémentés en utilisant des descripteurs de 4×4 histogrammes de 8 dimensions, soit $4 \times 4 \times 8 = 128$ éléments pour chaque descripteur.

FIG. 4.2 – Algorithme des SIFT

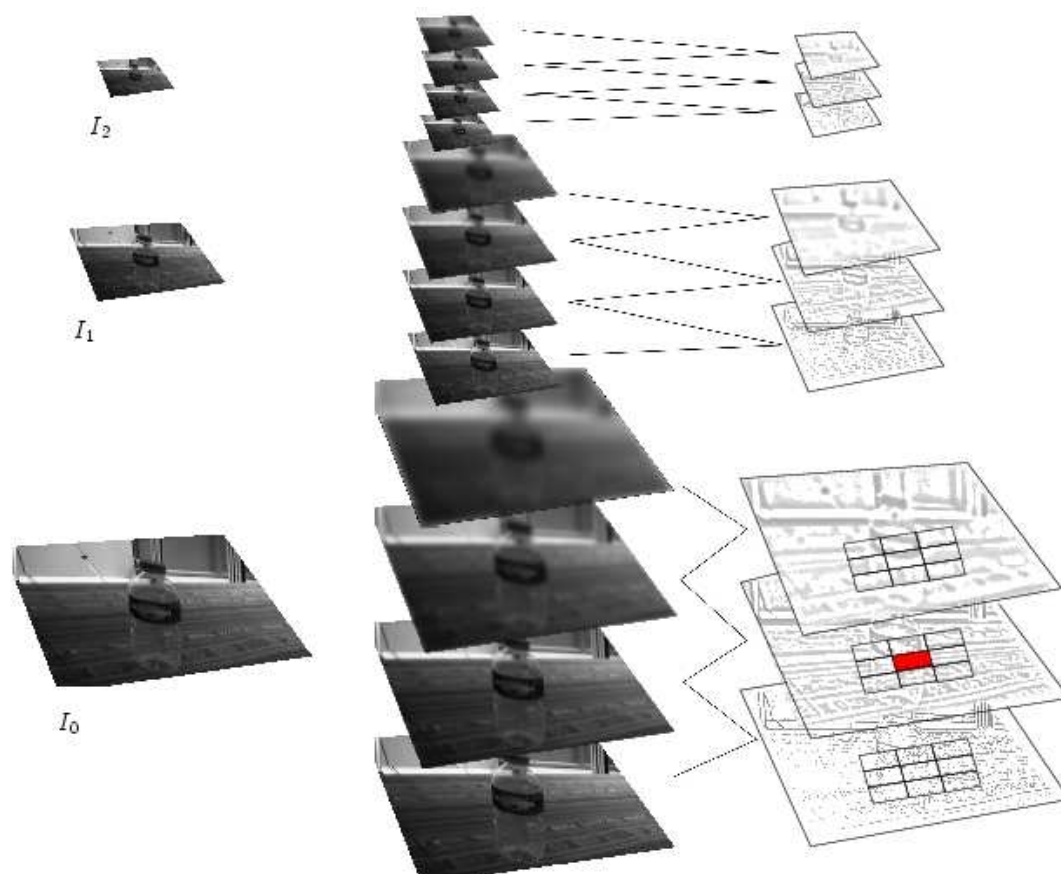


FIG. 4.3 – Détection des extremums dans l'espace d'échelle : une pyramide gaussienne est construite, puis pour chaque dimension, l'image est convoluée par des filtres gaussien de variance croissante. Les images résultantes sont soustraites deux à deux (DoG) donnant une approximation du Laplacien. Ensuite, chaque pixel est comparé aux huit pixels voisins à la même échelle et aux 9 pixels voisins dans les échelles adjacentes. Si sa valeur est maximum ou minimum, alors il est conservé comme candidat potentiel.

La mise en correspondance des descripteurs peut être basée sur le calcul de la distance euclidienne. Les points de l'image de référence sont comparés un à un à l'ensemble des points de l'image de test. La distance séparant deux points est calculée. Les mises en correspondance sont ensuite classées par ordre croissant de distance euclidienne. Le rapport entre la plus proche distance et la seconde plus proche distance est ensuite calculé. Plus ce rapport est proche de 1 et plus les distances sont proches. Cela signifie que la mise en correspondance n'est pas discriminante. Plus ce rapport diminue, plus l'écart entre les distances augmente et plus la mise en correspondance est correcte. En effet, cela signifie que le point de plus proche distance est similaire au point testé alors que tous les autres en sont éloignés. Finalement l'algorithme rejettera toutes les mises en correspondance qui résultent en un ratio supérieur à 0.8.

Le problème de l'identification de descripteurs similaires pour des vecteurs de grande dimension est la complexité élevée si la solution exacte est recherchée. La recherche *best-bin-first* (BBF) [Beis 97], basée sur les *k-d tree*, permet d'identifier efficacement les plus proches voisins avec une probabilité élevée. Pour améliorer la mise en correspondance, un poids double est attribué aux points qui sont générés à plus grande échelle.

4.2.3 Élimination des faux positifs.

La méthode de mise en correspondance, présentée dans le paragraphe précédent, permet de détecter les paires de descripteurs les plus pertinentes et d'éliminer une partie des faux positifs. Cependant, il peut arriver que plusieurs parties d'une image présentent les mêmes caractéristiques, par exemple si un même motif se répète. Il est alors possible d'éliminer les mauvaises paires en tenant compte de la position relative des points d'intérêt les uns par rapport aux autres.

Une première solution consiste à tenir compte de la position relative de points appartenant à un même objet, par exemple en utilisant la méthode de la transformée de Hough, comme le propose [Lowe 04].

Dans le cadre de notre étude, la contrainte épipolaire reliant deux images est connue à chaque instant et peut être utilisée pour vérifier la validité d'une mise en correspondance de deux points ${}^d\mathbf{x}$ et ${}^e\mathbf{x}$. Il suffit de montrer que la paire de points mis en correspondance vérifie la contrainte épipolaire (voir équation (2.18)), c'est-à-dire que chacun des points est porté par la droite épipolaire associée à l'autre point. La figure 4.4 illustre la méthode d'élimination des faux positifs. Dans la pratique, pour tenir compte du bruit de mesure de la position des points dans les images et des erreurs sur les positions des caméras, les points voisins de la ligne épipolaire seront conservés. Cette zone de tolérance est représentée en pointillés sur la figure 4.4.

4.3 Localisation d'un objet inconnu

Plusieurs vues sont nécessaires pour observer la portion atteignable de la ligne de vue (voir figure 4.5). Les descripteurs SIFT sont calculés pour chacune de ces vues et mis en correspondance avec ceux de la vue déportée, résultant en une liste, éventuellement vide, de points mis en correspondance. Dans la figure 4.5, l'objet est représenté par un rectangle vert et trois vues embarquées $e1$, $e2$ et $e3$ sont représentées. L'objet est inclus dans les deux premières vues. La troisième vue ne permet de détecter qu'un point de l'arrière plan. Dans cette section, un algorithme de localisation de l'objet à partir de ces informations est proposé. Il repose sur l'hypothèse que le plus grand nombre de mises en correspondance sera trouvé à l'emplacement de l'objet.

Pour mettre en commun les informations obtenues dans l'ensemble des vues, il faut les représenter dans un référentiel commun. Nous avons choisi de convertir les informations contenues dans les images en densité de probabilité représentant la position de l'objet le long de la ligne de vue. Le problème est ainsi ramené à un problème unidimensionnel d'estimation de la distance sur la ligne de vue.

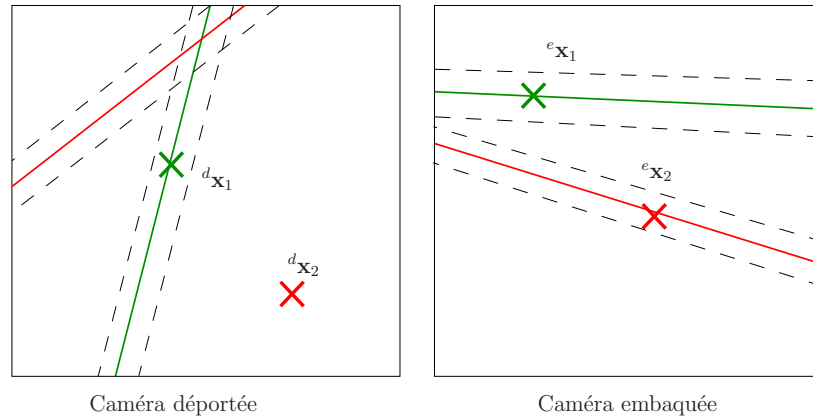


FIG. 4.4 – À gauche une image acquise depuis la caméra déportée, à droite, une image acquise depuis la caméra embarquée. Deux paires de points mis en correspondance sont représentées (${}^e\mathbf{x}_1$; ${}^d\mathbf{x}_1$) et (${}^e\mathbf{x}_2$; ${}^d\mathbf{x}_2$). La première paire vérifie la contrainte épipolaire tandis que la seconde paire est un faux positif

Aucune contrainte n'est imposée sur l'acquisition des images. Elles sont acquises à fréquence régulière le long de la ligne de vue en utilisant la loi de commande décrite dans le chapitre précédent. Deux images successives se chevauchent partiellement et permettent d'observer une portion commune de la ligne de vue. Cette superposition d'images peut entraîner une redondance dans les informations visuelles extraites. Il faut traiter cette redondance pour obtenir une bonne estimation de la position de l'objet.

L'algorithme de localisation d'un objet à partir d'un *clik* est présenté dans l'encadré 4.6 et sera détaillé dans les paragraphes à venir. Il comprend 3 étapes :

1. **Dans l'image déportée**, détecter les points d'intérêt \mathbf{x}_d au voisinage du *clik*.
2. **Dans l'image embarquée courante**, détecter les points d'intérêt \mathbf{x}_e , mettre en correspondance les K paires $(\mathbf{x}_d, \mathbf{x}_e)$ et calculer la vraisemblance associée à la position de l'objet pour les K paires, puis pour la vue.
3. **Fusionner** l'information courante à l'information précédemment obtenue.

La détection des points d'intérêt et leur mise en correspondance a été décrite dans le paragraphe précédent. Si la caméra déportée est fixe et la scène immobile, alors la détection des points d'intérêt au voisinage du *clik* peut être effectuée une fois pour toute en début de processus. Dans le cadre de cette étude, les points d'intérêt seront détectés sur une image déportée acquise simultanément à l'image embarquée courante, afin de tenir compte de changements de l'illumination ou d'un déplacement possible du support de la caméra déportée (typiquement si elle est fixée à un fauteuil roulant). Le paragraphe 4.3.1 décrit le calcul de la vraisemblance associée à une paire de points $(\mathbf{x}_d, \mathbf{x}_e)$ puis à une vue. Enfin, le paragraphe 4.3.2 décrit la fusion des informations sur l'ensemble des vues embarquées acquises lors du parcours de la ligne de vue.

4.3.1 Estimation de la vraisemblance associée à une vue

Pour chaque image acquise lors du parcours de la ligne de vue, connaissant la position relative des caméras, à chacune des paires de points 2D homologues ${}^d\mathbf{x}$ et ${}^e\mathbf{x}$ peut être associé un point 3D ${}^d\mathcal{X}$. Ce point 3D peut appartenir à l'objet et donner une information pertinente sur sa position sur la ligne de vue ou bien appartenir à l'arrière plan et donner une information fautive. Pour ne conserver que les informations pertinentes, une pondération sera attribuée à chacune des paires de points en fonction de la distance séparant ${}^d\mathcal{X}$ de la ligne de vue. Les contributions de l'ensemble des paires de points seront ensuite mises en commun pour former la vraisemblance sur la position de l'objet associée à une vue.

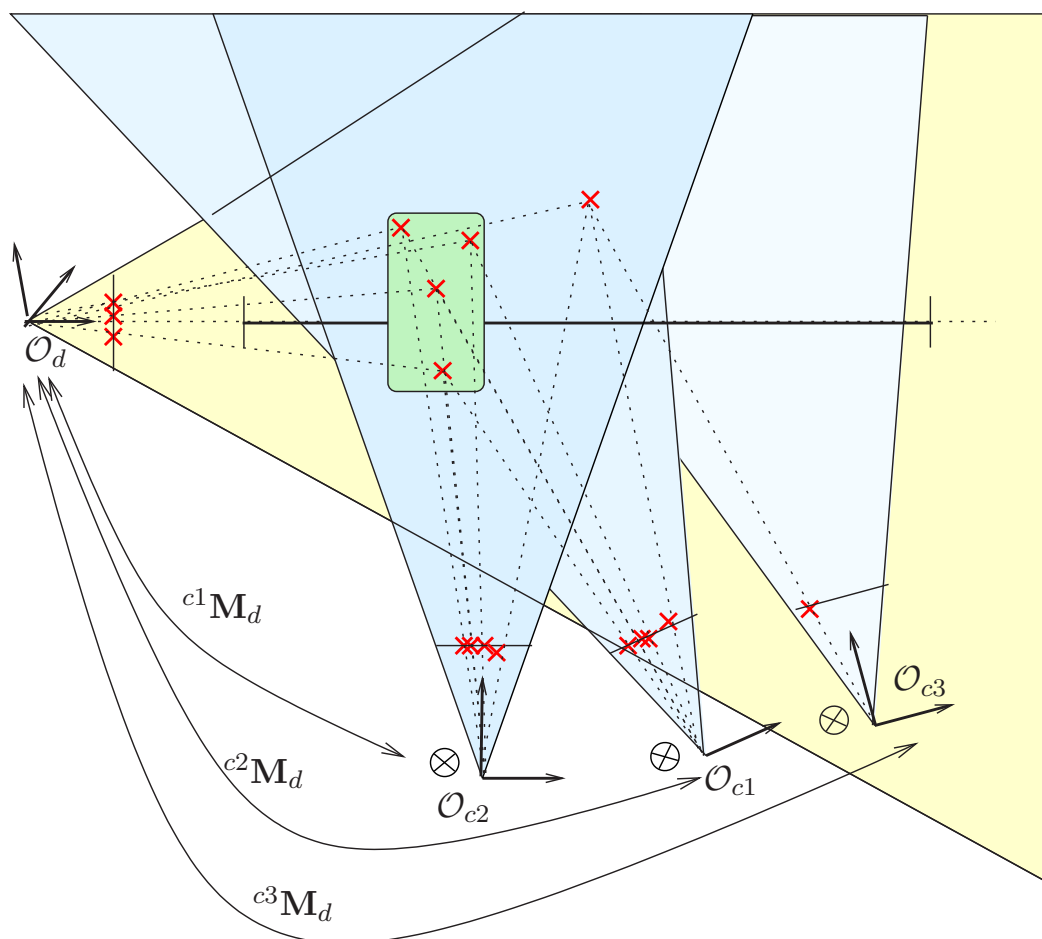


FIG. 4.5 – Localisation de l'objet le long de la ligne de vue. Dans cet exemple, trois vues embarquées sont acquises. L'objet est inclus dans les deux premières vues et absent de la troisième vue. Il est localisé à partir des mises en correspondance entre chacune de ces vues embarquées et la vue déportée.

One Click Focus : localisation d'un objet à partir d'un *clic*.

1. **Dans l'image déportée**, détecter les points d'intérêt \mathbf{x}_d au voisinage du *clic*.
2. **Dans chaque image embarquée**, détecter les points d'intérêt et mettre en correspondance les K paires $(\mathbf{x}_d, \mathbf{x}_e)$,
 - (a) **Pour chaque paire de points**
 - i. Estimer la position du point 3D ${}^d\mathbf{X}$ par triangulation.
 - ii. Calculer la projection orthogonale du point ${}^d\mathbf{X}$ sur la ligne de vue. Elle est notée D .
 - iii. Calculer la distance euclidienne séparant le point ${}^d\mathbf{X}$ de la ligne de vue. Cette distance est notée h .
 - iv. Calculer l'incertitude sur la mesure de ce point en fonction de la position courante de la caméra embarquée dans le repère de la caméra déportée et de la position des points \mathbf{x}_e et \mathbf{x}_d . Elle est notée σ_D
 - v. Calculer la vraisemblance associée au point : c'est une gaussienne centrée sur D , d'amplitude $1/h$ et de variance σ_D .
 - (b) **Pour les K paires de points détectées**, calculer la vraisemblance de la position de l'objet.
3. **Sur l'ensemble des vues**, fusionner les densités de probabilités calculées pour chaque image embarquée :
 - (a) Compter le nombre de vues dans lesquelles apparaissent chaque portion de la ligne de vue
 - (b) Calculer la densité de probabilité associée à l'ensemble des vues

FIG. 4.6 – *Algorithme de localisation d'un objet inconnu à partir d'un clic.*

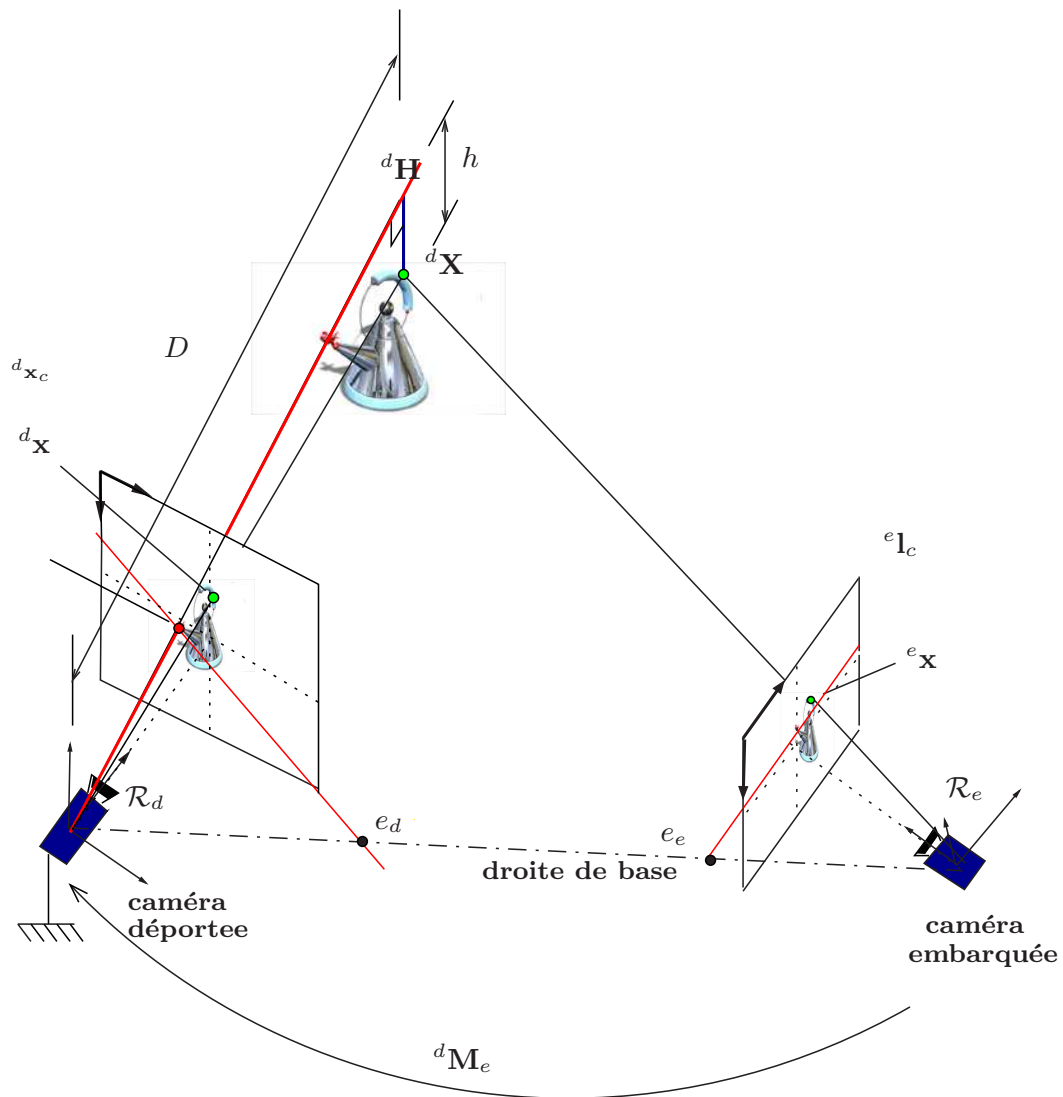


FIG. 4.7 – À partir d'une paire de points homologues et connaissant la position relative des deux caméras, le point 3D $^d\mathbf{X}$ correspondant peut être calculé. Sa projection orthogonale $^d\mathbf{H}$ sur la ligne de vue associée au point cliqué donne une indication sur la profondeur D de ce point le long de la ligne de vue. La distance h est la distance euclidienne entre le point et la ligne de vue.

La figure 4.7 illustre l'algorithme permettant de calculer la contribution d'une mise en correspondance à l'estimation de la profondeur. Le calcul de la vraisemblance associée à une mise en correspondance nécessite le calcul de quatre valeurs :

1. Estimer la position du point 3D ${}^d\mathbf{X}$ par triangulation : 4.3.1.1.
2. Calculer la distance séparant la centre optique de la caméra déportée et la projection orthogonale du point ${}^d\mathbf{X}$ sur la ligne de vue. Elle est notée D : 4.3.1.2.
3. Calculer la distance euclidienne séparant le point ${}^d\mathbf{X}$ de la ligne de vue. Cette distance est notée h : 4.3.1.2.
4. Calculer l'incertitude sur la mesure de ce point en fonction de la position courante de la caméra embarquée dans le repère de la caméra déportée et de la position des points \mathbf{x}_e et \mathbf{x}_d . Elle est notée σ_D : 4.3.1.3.

La vraisemblance associée à une mise en correspondance est une fonction gaussienne centrée sur D , d'amplitude $A = \frac{1}{\sigma_h \sqrt{2\pi}} e^{-\frac{h^2}{2\sigma_h^2}}$ et de variance σ_D (voir figure 4.8).

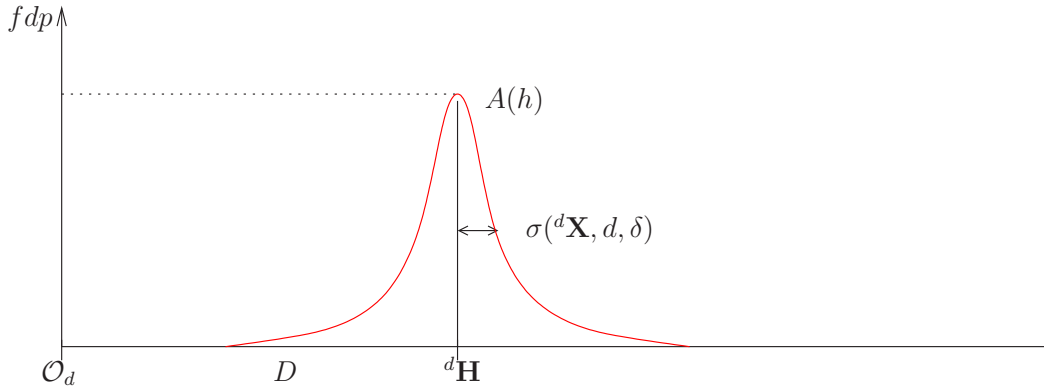


FIG. 4.8 – Construction de la fonction gaussienne attachée à un point. Son amplitude A est fonction de sa distance à la ligne de vue. Sa variance σ^2 représente l'incertitude sur la position. Elle est liée à sa distance au centre de projection \mathcal{O}_d , à la disparité D de la mesure des points homologues dans l'image et à l'erreur de mesure de la position relative des deux caméras.

Cette représentation sera ensuite utilisée pour fusionner les contributions des points mis en correspondance pour une vue embarquée, puis pour fusionner les informations sur l'ensemble des images acquises tout au long de la ligne de vue.

4.3.1.1 Calcul des coordonnées du point 3D ${}^d\mathbf{X}$

Considérons une paire de points 2D homologues ${}^d\mathbf{x}({}^d x, {}^d y, 1)^\top$ et ${}^e\mathbf{x}({}^e x, {}^e y, 1)^\top$ exprimés par leurs coordonnées métriques dans les images de la caméra déportée et de la caméra embarquée. ${}^d\mathcal{X}$ est le point 3D correspondant, exprimé dans le repère de la caméra déportée \mathcal{R}_d .

Soit \mathbf{P} , la matrice de projection, les projections métriques du point ${}^d\mathcal{X}$ sur les plans images des deux caméras s'écrivent :

$$\begin{aligned} {}^d\mathbf{x} &= \mathbf{P}^d \mathbf{X} \\ {}^e\mathbf{x} &= \mathbf{P}^e \mathbf{M}_d {}^d\mathbf{X} \end{aligned}$$

En développant la première équation, l'expression des coordonnées du point ${}^d\mathbf{x}$ s'écrit :

$$\begin{cases} {}^d x &= \frac{{}^d X}{{}^d Z} \\ {}^d y &= \frac{{}^d Y}{{}^d Z} \end{cases} \quad (4.1)$$

et à partir de l'écriture suivante de la matrice ${}^e \mathbf{M}_d$:

$${}^e \mathbf{M}_d = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

l'expression des coordonnées de ${}^e \mathbf{x}$ peut être déduite :

$$\begin{cases} {}^e x &= \frac{{}^e X}{{}^e Z} = \frac{r_{11} {}^d X + r_{12} {}^d Y + r_{13} {}^d Z + t_x}{r_{31} {}^d X + r_{32} {}^d Y + r_{33} {}^d Z + t_z} \\ {}^e y &= \frac{{}^e Y}{{}^e Z} = \frac{r_{21} {}^d X + r_{22} {}^d Y + r_{23} {}^d Z + t_y}{r_{31} {}^d X + r_{32} {}^d Y + r_{33} {}^d Z + t_z} \end{cases} \quad (4.2)$$

En introduisant (4.1) dans (4.2) et en notant \mathbf{r}_i la i ème ligne de la matrice de rotation, l'expression devient :

$$\begin{cases} {}^e x &= \frac{{}^d Z \mathbf{r}_1^\top {}^d \mathbf{x} + t_x}{{}^d Z \mathbf{r}_3^\top {}^d \mathbf{x} + t_z} \\ {}^e y &= \frac{{}^d Z \mathbf{r}_2^\top {}^d \mathbf{x} + t_y}{{}^d Z \mathbf{r}_3^\top {}^d \mathbf{x} + t_z} \end{cases}$$

Le système est alors un système de deux équations à une seule inconnue ${}^d Z$. Pour tenir compte du bruit dans la mesure, l'expression de la coordonnée ${}^d Z$ est la moyenne des solutions apportées par chacune de ces équations :

$${}^d Z = \frac{1}{2} \left(\frac{t_x - {}^e x t_z}{(\mathbf{r}_3^\top {}^d \mathbf{x}) {}^e x - \mathbf{r}_1^\top {}^d \mathbf{x}} + \frac{t_y - {}^e y t_z}{(\mathbf{r}_3^\top {}^d \mathbf{x}) {}^e y - \mathbf{r}_2^\top {}^d \mathbf{x}} \right) \quad (4.3)$$

Ainsi, à partir des équations (4.1) et (4.3), on obtient une estimation des coordonnées 3D ${}^d \bar{\mathbf{X}}({}^d X, {}^d Y, {}^d Z)^\top$. L'étape suivante consiste à projeter ce point sur la ligne de vue.

4.3.1.2 Projection orthogonale sur la ligne de vue.

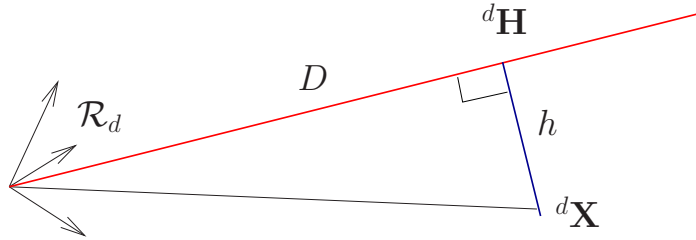


FIG. 4.9 – La distance h est la distance séparant le point de la ligne de vue (voir 4.7).

La projection du point ${}^d \mathcal{X}$ sur la ligne de vue est le produit scalaire, dans l'espace euclidien, du vecteur correspondant au point ${}^d \mathcal{X}$ et du vecteur directeur de la ligne de vue associée au clic \mathbf{w} , qui peut être déduit de sa position 3D ${}^d \mathbf{x}_{clic}(x_{clic}, y_{clic}, 1)^\top$ dans le repère de la caméra déportée :

$$\mathbf{w} = \frac{{}^d \mathbf{x}_{clic}}{|{}^d \mathbf{x}_{clic}|}$$

En notant ${}^d\mathbf{x}({}^dx, {}^dy, 1)^\top$, la position de ${}^d\mathbf{x}$ dans \mathcal{R}_d , la distance séparant le centre optique de la caméra déportée de la projection orthogonale de ${}^d\bar{\mathbf{X}}$ sur la ligne de vue est :

$$D = {}^d\bar{\mathbf{X}}^\top \mathbf{w} = {}^dZ({}^d\mathbf{x}^\top \mathbf{w})$$

D dépend de dZ (voir équation (4.3)) et des positions du point considéré ${}^d\mathbf{x}$ et du clic ${}^d\mathbf{x}_{clic}$ dans l'image de la caméra déportée :

$$D = \frac{{}^dZ}{|{}^d\mathbf{x}_{clic}|} ({}^d\mathbf{x}^\top {}^d\mathbf{x}_{clic}) \quad (4.4)$$

et le projeté orthogonal de ${}^d\bar{\mathbf{X}}$ sur la ligne de vue est le point :

$${}^d\mathbf{H} = D\mathbf{w}$$

Le triangle $({}^d\mathcal{O}, {}^d\mathbf{H}, {}^d\bar{\mathbf{X}})$ étant rectangle en ${}^d\mathbf{H}$, la distance h séparant le point ${}^d\bar{\mathbf{X}}$ de sa projection ${}^d\mathbf{H}$ sur la ligne de vue est donc immédiatement déduite des équations précédentes :

$$h = |{}^d\bar{\mathbf{X}} - {}^d\mathbf{H}|$$

4.3.1.3 Calcul de l'incertitude

Parmi les hypothèses émises sur le système (voir annexe C), il est admis que la position relative des deux caméras peut être mesurée à chaque instant.

En pratique, la transformation ${}^d\mathbf{M}_e$ mesurée n'est pas exacte car elle est soumise aux imprécisions de l'étalonnage du système de vision et au bruit de mesure de la position articulaire du robot. Le calcul des coordonnées des points homologues ${}^d\mathbf{x}$ et ${}^e\mathbf{x}$ n'est pas non plus exact car il est issu d'un processus de traitement de l'image qui est potentiellement bruité. Ces erreurs de mesure se répercutent sur la localisation du point 3D, sa projection sur la ligne de vue et donc sur la distance associée. Par exemple, plus un point est éloigné du système de vision, plus le calcul de sa position 3D est sensible aux erreurs de mesure (voir figure 4.10). L'incertitude résultante des erreurs de mesure peut être évaluée. Elle dépend des mesures courantes et des variances associées à chacun des paramètres.

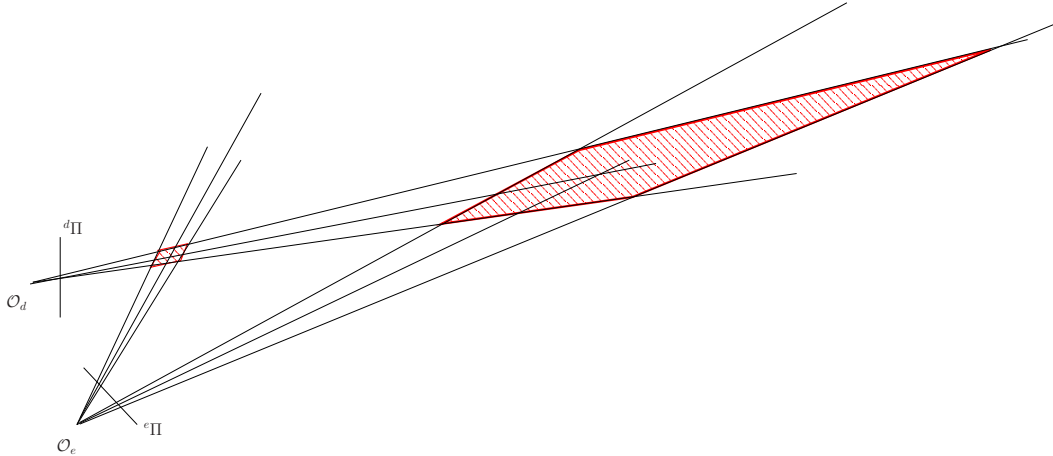


FIG. 4.10 – Plus le point 3D est distant des centres de projection des caméras, plus une erreur de mesure entraîne une erreur de localisation importante, plus la variance attachée au point est importante.

On souhaite établir l'impact qu'ont sur l'estimation de la position 3D du point ${}^d\mathbf{X}$, les erreurs de mesure de la position des points 2D formant une paire $(\mathbf{x}_e, \mathbf{x}_d)$ et l'erreur de mesure de la pose

relative des caméras. On construit un vecteur η qui est un vecteur d'état représentant l'ensemble des paramètres du système dont on souhaite étudier l'impact. Soit \mathbf{P} la matrice de covariance associée :

$$\eta = \begin{pmatrix} t_x \\ t_y \\ t_z \\ \theta_x \\ \theta_y \\ \theta_z \\ e_x \\ e_y \\ d_x \\ d_y \end{pmatrix} \quad \text{avec} \quad \mathbf{P} = \begin{pmatrix} P_{t_x t_x} & P_{t_x t_y} & P_{t_x t_z} & \dots \\ P_{t_y t_x} & P_{t_y t_y} & P_{t_y t_z} & \dots \\ P_{t_z t_x} & P_{t_z t_y} & P_{t_z t_z} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (4.5)$$

Les éléments du vecteur d'état peuvent être supposés indépendants. En pratique, ce n'est pas tout à fait le cas. La position relative des deux caméras est calculée à partir de la mesure courante de la position de l'effecteur ${}^b\mathbf{M}_g$, de la position relative de la caméra embarquée et de l'effecteur ${}^g\mathbf{M}_e$ et de la position de la caméra déportée ${}^b\mathbf{M}_e$. ${}^g\mathbf{M}_e$ et ${}^b\mathbf{M}_e$ présentent des corrélations, en particulier, les rotations autour de \mathbf{i} et \mathbf{j} sont corrélées aux translations le long de ces deux axes. Ainsi, en supposant les paramètres étudiés indépendants, les termes de covariance peuvent être négligés devant les variances des éléments du vecteur d'état, la matrice de covariance est donc diagonale. L'incertitude associée à la projection d'un point sur la ligne de vue s'exprime donc de la manière suivante :

$$\begin{aligned} \sigma_D^2 &= \text{var}(D) \\ &= \frac{\partial D}{\partial \eta} \mathbf{P} \frac{\partial D}{\partial \eta}^\top \\ &= \mathbf{J} \mathbf{P} \mathbf{J}^\top \\ &= P_{t_x t_x} \left(\frac{\partial D}{\partial t_x} \right)^2 + P_{t_y t_y} \left(\frac{\partial D}{\partial t_y} \right)^2 + P_{t_z t_z} \left(\frac{\partial D}{\partial t_z} \right)^2 + \\ &\quad P_{\theta_x \theta_x} \left(\frac{\partial D}{\partial \theta_x} \right)^2 + P_{\theta_y \theta_y} \left(\frac{\partial D}{\partial \theta_y} \right)^2 + P_{\theta_z \theta_z} \left(\frac{\partial D}{\partial \theta_z} \right)^2 + \\ &\quad P_{e_x e_x} \left(\frac{\partial D}{\partial e_x} \right)^2 + P_{e_y e_y} \left(\frac{\partial D}{\partial e_y} \right)^2 + P_{d_x d_x} \left(\frac{\partial D}{\partial d_x} \right)^2 + P_{d_y d_y} \left(\frac{\partial D}{\partial d_y} \right)^2 \end{aligned} \quad (4.6)$$

La figure 4.11 illustre l'évolution de la variance associée au calcul de la profondeur d'un point 3D pour différentes positions du point et différentes configurations des caméras. Comme attendu, plus le point 3D est éloigné du système de vision, plus la variance augmente. La position relative des caméras joue aussi un rôle ; lorsque la ligne de base du système de stéréovision est petite, la variance est importante. Les variances les plus faibles sont obtenues lorsque les caméras sont relativement éloignées et leurs plans images sont perpendiculaires. Les meilleurs résultats sont obtenus dans la configuration ${}^d\mathbf{M}_{e4}$ et ${}^d\mathbf{M}_{e5}$. Le calcul de cette incertitude permet de fixer l'écart-type de la fonction gaussienne, c'est-à-dire à construire la vraisemblance associée aux coordonnées d'un point.

4.3.1.4 Construction de la vraisemblance associée au point

Pour tenir compte de l'incertitude engendrée par les erreurs de mesure, la projection d'un point sur la ligne de vue est représentée par une fonction gaussienne dont la moyenne est la distance D , l'amplitude A est relative à l'inverse de la distance h séparant le point de la ligne de vue et sa variance est la variance calculée dans l'équation (4.6) (voir figure(4.8)).

$$\mathcal{G} = \mathcal{G}(A, D, \sigma_D, D) = \frac{A}{\sigma_D \sqrt{2\pi}} e^{-\left(\frac{D-x}{2\sigma_D}\right)^2} \quad \text{avec} \quad A = \frac{1}{\sigma_h \sqrt{2\pi}} e^{-\frac{h^2}{2\sigma_h^2}} \quad (4.7)$$

Cette représentation permet de pondérer la contribution de chaque mise en correspondance en fonction de sa distance à la ligne de vue. Ainsi, la contribution d'un point éloigné, qui appartient certainement au fond, est faible relativement à la contribution d'un point proche de la ligne de vue qui appartient vraisemblablement à l'objet.

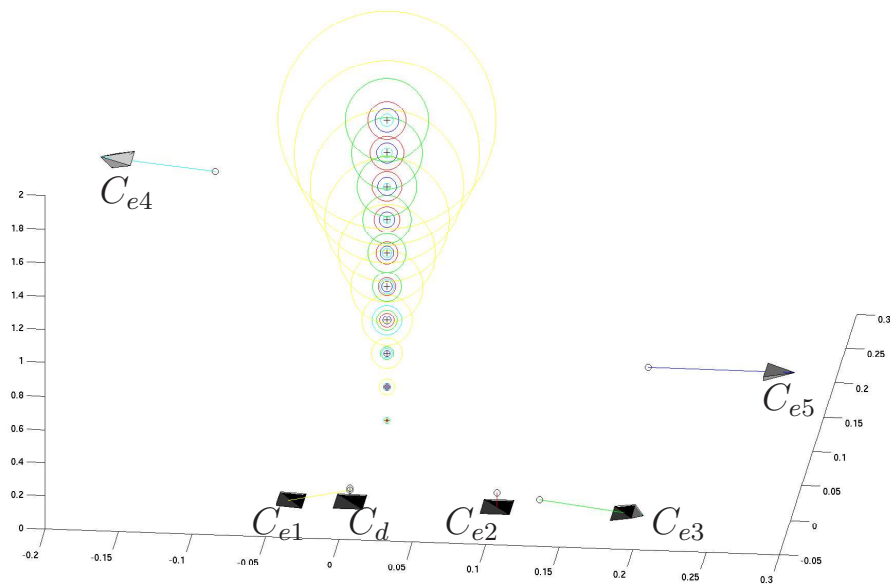


FIG. 4.11 – Incertitude en fonction de la configuration des caméras et de la position du point 3D. Les cercles jaunes représentent les variances associées à la caméra C_{e1} , les cercles rouges sont associés à C_{e2} , les cercles verts à C_{e3} , les cercles de couleur cyan à C_{e4} et bleus à C_{e5} .

4.3.1.5 Fusion des informations sur une vue

Chacune des images acquises le long de la ligne de vue ne permet d'observer qu'une portion de la ligne de vue (voir la figure 4.12). Pour chaque portion de la droite visualisée, un ensemble de mises en correspondances est réalisé, donnant un ensemble de fonctions gaussiennes.

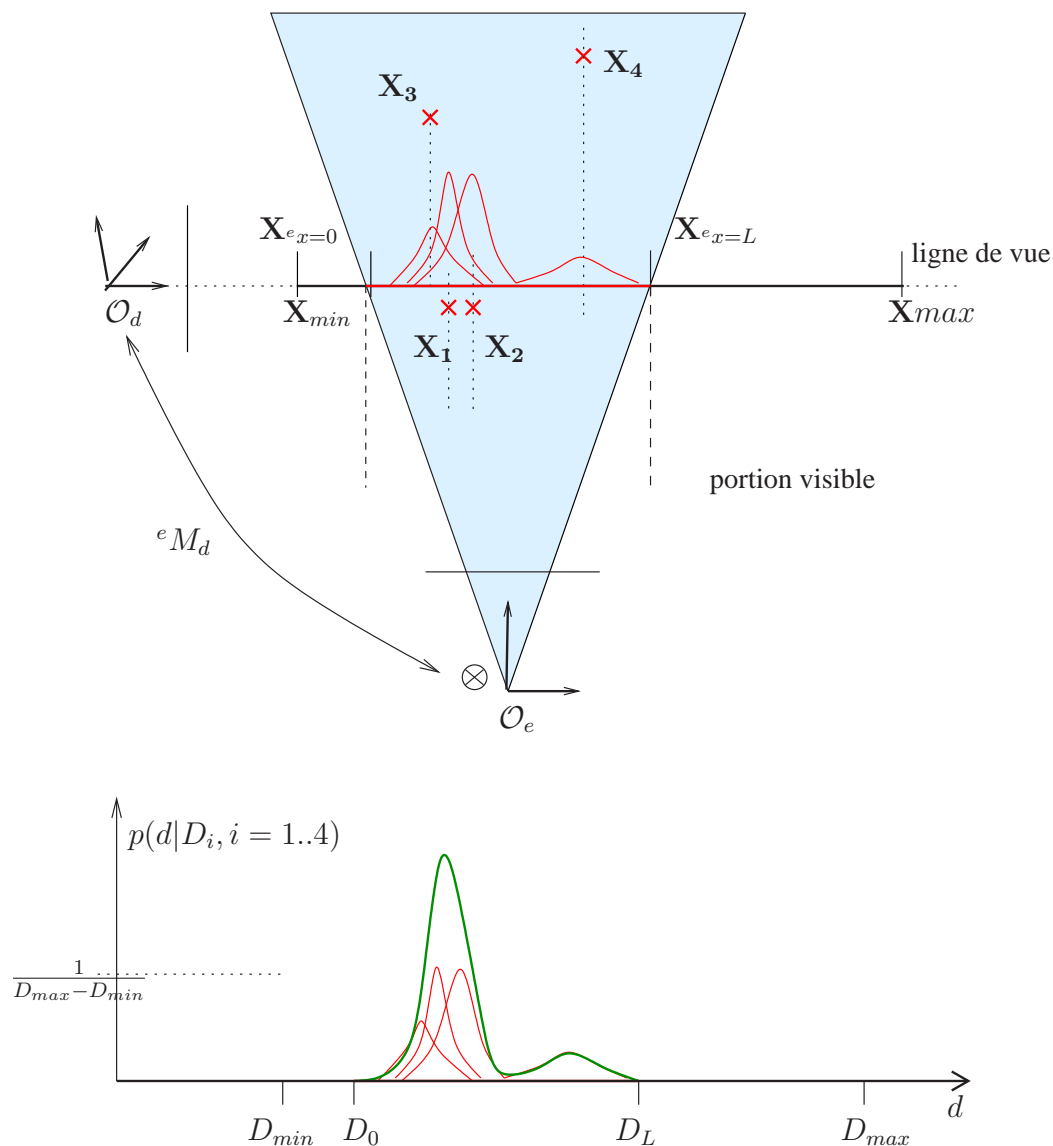


FIG. 4.12 – Mélange de gaussiennes sur la portion de la ligne de vue visible (en vert). En rouge les gaussiennes associées à chacun des points P_i

La probabilité associée à la distance de l'objet sur la ligne de vue (D) connaissant la projection d'un ensemble de point 3D peut être exprimée par un mélange de gaussiennes :

$$p(x) = \frac{1}{N} \sum_{i=1}^K \mathcal{G}(A_i, D_i, \sigma_{D_i}, x) \quad (4.8)$$

Le terme N est un facteur de normalisation construit de façon à assurer l'égalité :

$$\int_0^{+\infty} p(x)dx = 1 \quad (4.9)$$

Pour K points observés, l'équation (4.9) devient

$$\begin{aligned} \int_{-\infty}^{+\infty} p(x)dx &= 1 \\ &= \int_{-\infty}^{+\infty} \frac{1}{N} \sum_{i=1}^K \mathcal{G}(A_i, D_i, \sigma_{D_i}, x) dx \\ &= \frac{1}{N} \sum_{i=1}^K \int_{-\infty}^{+\infty} \mathcal{G}(A_i, D_i, \sigma_{D_i}, x) dx \\ &= \frac{1}{N} \sum_{i=1}^K A_i \end{aligned}$$

d'où :

$$N = \frac{1}{\sum_{i=1}^K A_i} \quad (4.10)$$

Le facteur de normalisation est obtenu en sommant les amplitudes des gaussiennes associées aux points mis en correspondance. Une fonction de densité de probabilité sur la distance à l'objet peut être associée à chacune des vues acquises le long du parcours de la ligne épipolaire.

4.3.2 Fusion des données sur l'ensemble des vues.

Pour fusionner les informations sur plusieurs vues, une première solution consiste à effectuer un mélange de gaussiennes sur l'ensemble des points détectés dans l'ensemble des vues, exactement de la même manière que dans le cas d'une seule vue.

Cependant, les images successives peuvent se chevaucher et donner des informations redondantes, en particulier lorsque la vitesse de déplacement diminue à l'approche d'une extrémité du segment. La figure 4.13 illustre un cas pathologique dans lequel des faux positifs sont détectés dans trois vues successives alors que les points appartenant réellement à l'objet n'ont été vus que dans une image. La méthode de fusion des informations doit permettre de résoudre un tel cas en ne comptant qu'une seule fois un même point.

Pour résoudre ce problème, une *fonction de visibilité* est construite et notée $V = \sum_t v(x, t)$, où x est l'abscisse sur la ligne de vue et t le temps. Elle permet de compter le nombre de fois où une même partie du segment est vue. v est défini de la manière suivante :

$$v(d, t) = \begin{cases} 1 & \text{si } D_0(t) \leq D \leq D_L(t) \\ 0 & \text{sinon} \end{cases} \quad (4.11)$$

où D_0 et D_L sont définies à l'intersection du champ de vision de la caméra embarquée avec la ligne de vue. Lors de l'acquisition des images, la ligne de vue se projette horizontalement et est centrée dans la vue de la caméra embarquée. Le repère attaché à la caméra embarquée \mathcal{R}_e est donc positionné tel que le vecteur directeur \mathbf{k} est dans le plan épipolaire et le vecteur \mathbf{j} est normal au plan épipolaire. D_0 et D_L sont donc obtenues par triangulation du clic avec les points ${}^e\mathbf{x}_{p_0}(0, 0, 1)^\top$ et ${}^e\mathbf{x}_{p_L}(L, 0, 1)^\top$ de l'image de la caméra embarquée, où L est la largeur de l'image (voir équation (4.3)).

Pour I images acquises et $K = \sum_{i=0}^I K_i$ points observés, la fonction de densité de probabilité sur D s'écrit donc :

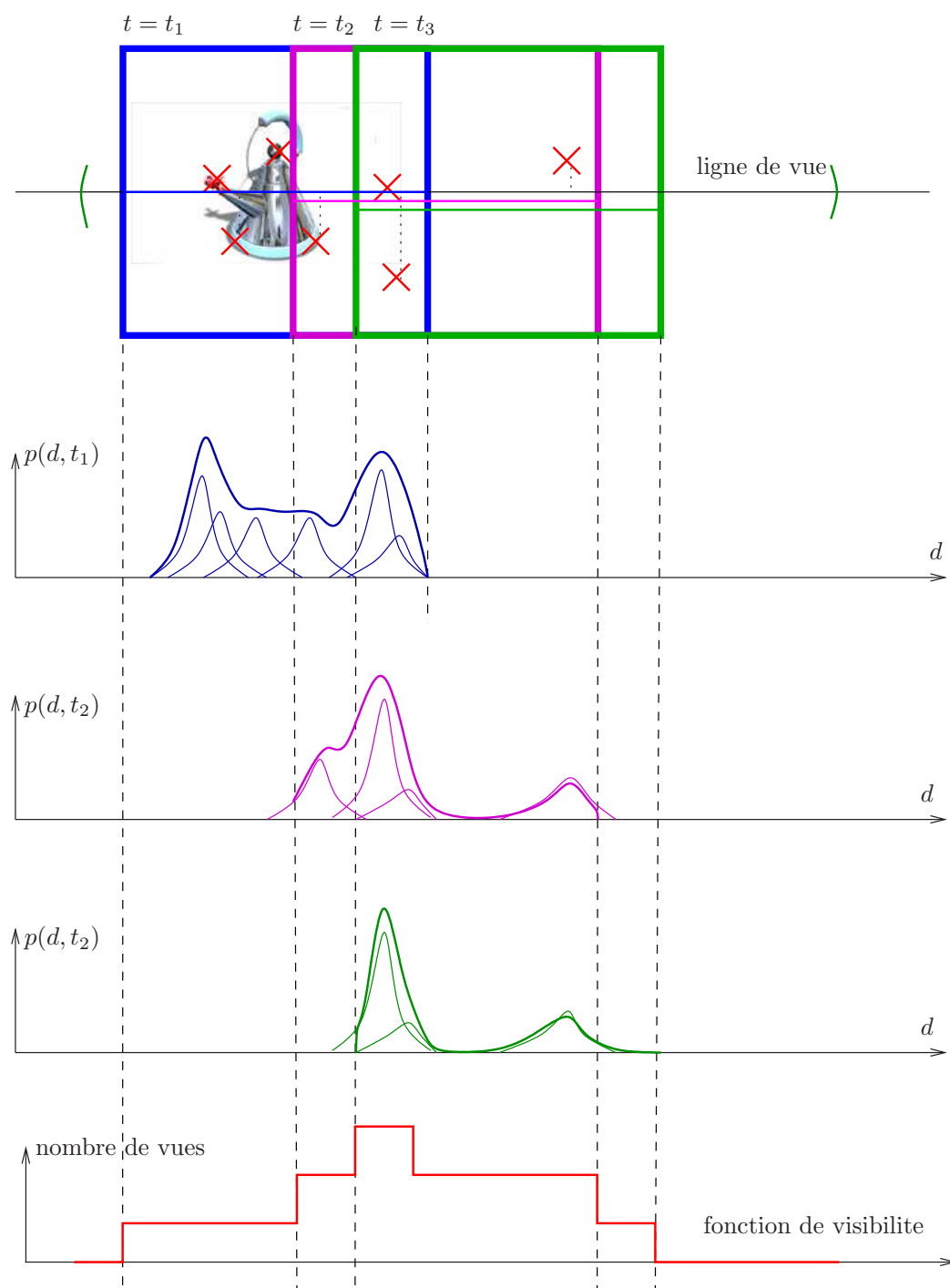


FIG. 4.13 – Acquisition de trois vues successives dans lesquelles apparaissent les mises en correspondance de paires identiques. La normalisation du mélange de gaussiennes sur ces trois vues permet de ne pas surévaluer la contribution d'une même paire de points détectée plusieurs fois. Dans cet exemple, si les résultats ne sont pas normés, la localisation de l'objet est fautive car deux faux positifs sont détectés dans les trois vues.

$$p(x) = \frac{1}{V} \sum_{i=1}^I \left(\frac{1}{N_i} \sum_{k=1}^{K_i} \mathcal{G}(A_i, D_i, \sigma_{D_i}, x) \right) = \frac{1}{V} \frac{1}{N} \sum_{k=1}^K \mathcal{G}(A_i, D_i, \sigma_{D_i}, x) \quad (4.12)$$

Finalement, cela revient à effectuer la moyenne des mesures sur plusieurs vues : si un même point est détecté plusieurs fois dans une séquence d'images, sa contribution est normée par le nombre de vues dans lesquelles il apparaît.

Le paragraphe suivant présente les résultats obtenus sur le robot cartésien, AFMA6, à l'IRISA, et sur le robot anthropomorphe, RX90, au CEA.

4.4 Résultats expérimentaux

Reprenons l'exemple donné en introduction de cette partie par la figure 4.1. Le robot utilisé est le robot RX90 et la scène est composée d'une table recouverte d'une nappe texturée sur laquelle un objet est posé, à portée du bras. À l'instant initial, l'objet est dans la vue de la caméra déportée et l'utilisateur le sélectionne en un *clac*. La position du bras est quelconque et l'objet n'est pas inclus *a priori* dans la vue de la caméra embarquée. La localisation peut être décomposée en quatre phases :

1. La sélection de l'objet par un *clac*
2. Une étape de centrage de la ligne de vue
3. Une étape de parcours de la ligne de vue, et la localisation de l'objet
4. Une étape de centrage et de zoom sur l'objet localisé

Les points SIFT sont extraits dans les images de la caméra embarquée lorsque la tâche de centrage de la ligne de vue est réglée. La figure 4.14 présentent les mises en correspondances obtenues au cours de la localisation de la bouteille. La dernière vue de la caméra embarquée ne contient pas l'objet, cependant des faux positifs sont détectés. Si leurs positions respectent la contrainte épipolaire, alors ils sont conservés, sinon ils sont rejetés. D'autres part, leur contributions est inversement proportionnelle à la distance 3D qui les séparent de la ligne de vue. Ainsi, ces points ont une pondération inférieure à ceux détectés dans la vue précédente.

La figure 4.15 présente le résultat de localisation de l'objet. Les points d'intérêt sont détectés sur la texture avoisinant la bouteille transparente. L'expérience 4.16 montre que même si le fond est uniforme, un objet transparent peut être localisé. Ceci est vrai tant qu'il y a suffisamment de texture dans le motif sélectionné dans la vue de la caméra déportée pour que ce motif puisse être identifié dans la vue de la caméra embarquée en utilisant une mise en correspondance de points SIFT.

En fin de processus, la localisation estimée de l'objet, c'est-à-dire un point 3D de coordonnées \mathbf{X}_o , est centré dans la vue de la caméra embarquée en utilisant un asservissement identique à celui permettant le centrage des extrémités du segment 3D. Dans le cadre de ces expériences, la précision de la localisation n'a pas été quantifiée. L'expérience est considérée réussie si la tâche de centrage sur la localisation 3D permet effectivement d'amener l'objet d'intérêt au centre de la vue embarquée.

Les figures 4.16 et 4.17 présentent d'autres résultats de localisation obtenus sur le RX90 du CEA-LIST pour la bouteille, placée cette fois sur un fond uniforme et un objet partiellement occulté sous un tas d'objets divers.

Les figure 4.18 et 4.19 présentent un résultat obtenu sur le robot AFMA6 de l'INRIA Rennes dans lequel plusieurs objets sont sélectionnés successivement. Dans cet exemple, la scène est composée d'un ensemble d'objets sur un fond texturé et plan. Les objets sont immobiles au cours du processus de localisation. Quatre objets sont tour à tour sélectionnés. À l'instant initial, la caméra embarquée est orientée vers le plafond. On peut voir un spot dans l'image. Cette vue est très éloignée de la vue désirée qui doit contenir l'objet sélectionné, c'est-à-dire le téléphone. Ensuite, le parcours nécessaire pour centrer successivement les trois autres objets est plus court.

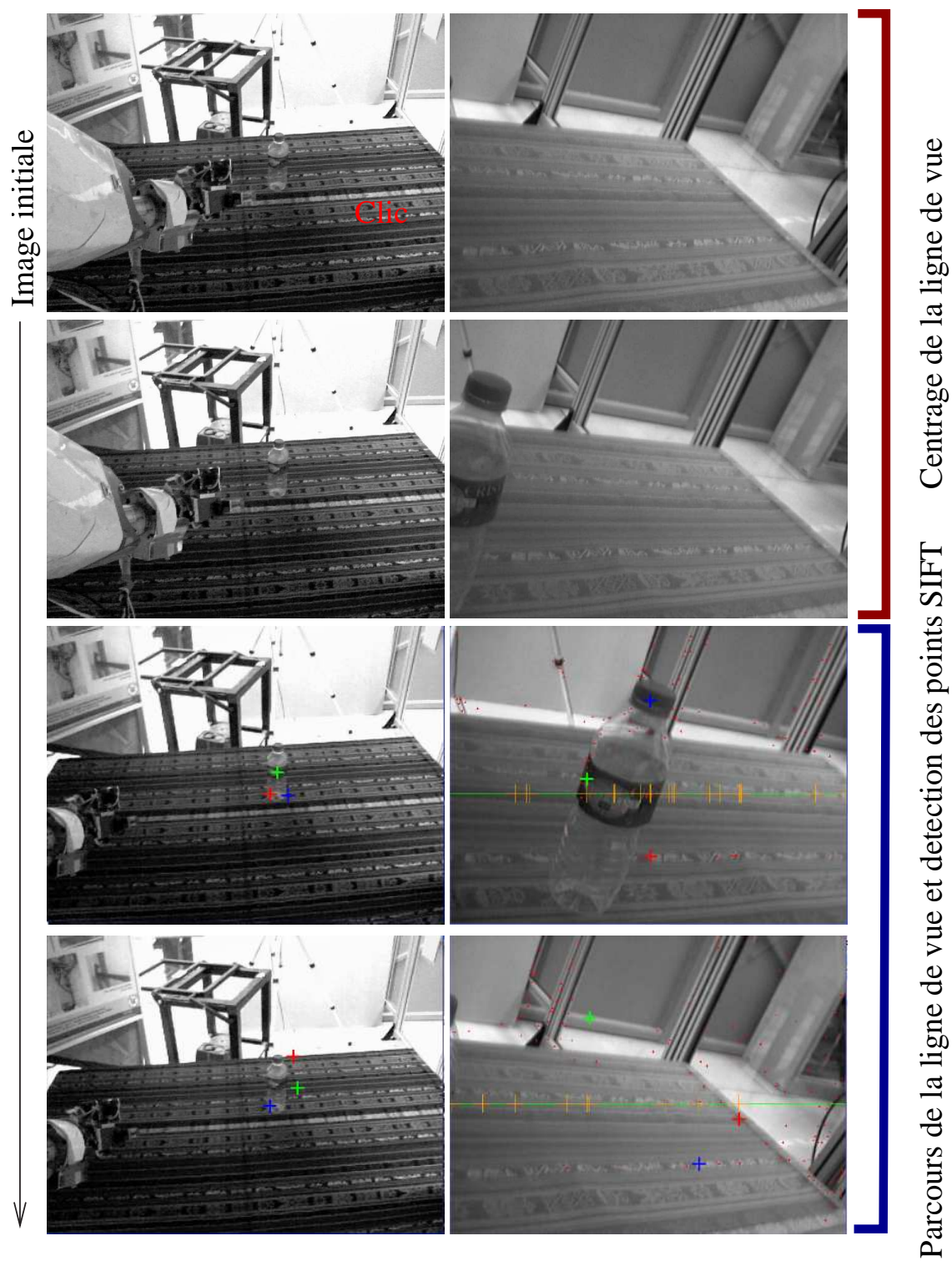


FIG. 4.14 – Détection et mise en correspondance de points SIFT au cours du parcours de la ligne de vue.

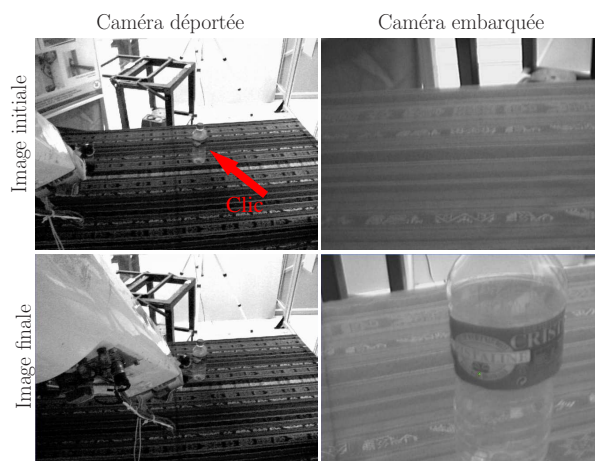


FIG. 4.15 – Localisation d'une bouteille d'eau sur un fond texturé sur le RX90.

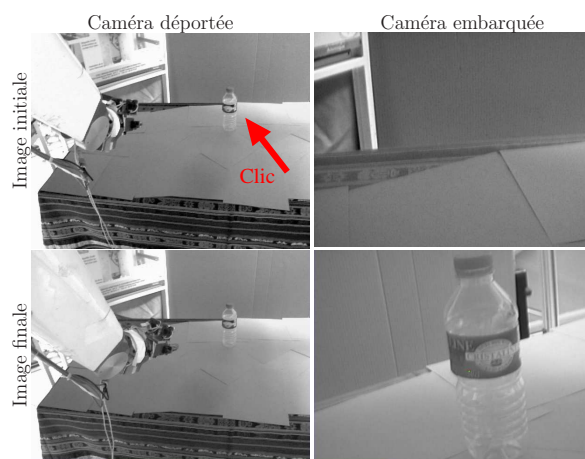


FIG. 4.16 – Localisation d'une bouteille d'eau sur un fond uniforme en utilisant le RX90.

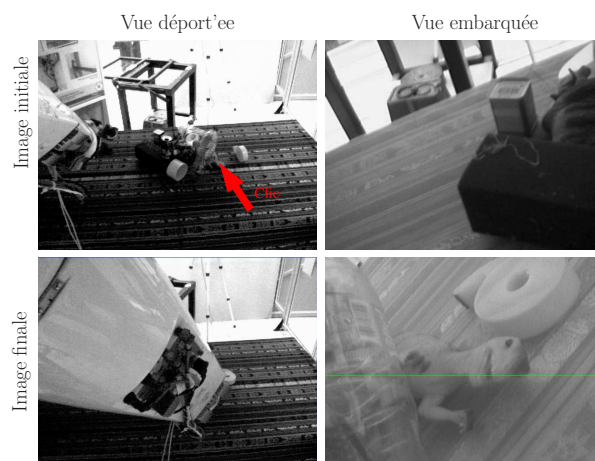


FIG. 4.17 – Localisation d'un jouet dans une scène complexe.

La recherche de l'objet débute lorsque la ligne de vue est centrée dans l'image déportée. Les points SIFT sont alors détectés dans l'image courante et mis en correspondance avec ceux de l'image déportée. Au fur et à mesure de l'acquisition des vues sur le parcours de la ligne épipolaire, les informations sont fusionnées pour construire la vraisemblance de la position de l'objet. Lorsque l'objet est localisé, une dernière phase consiste à centrer l'objet localisé dans l'image de la caméra embarquée.

La figure 4.19 détaille l'évolution de la vraisemblance de la position de l'objet au cours du processus de localisation. Selon l'orientation de la caméra embarquée relativement à la position de la ligne de vue portant l'objet, une ou plusieurs vues sont nécessaires pour visualiser le segment d'intérêt, ici entre $0.5m$ et $2,5m$:

- 3 vues pour localiser le téléphone ;
- 2 vues pour localiser le masque ;
- 1 vue pour localiser la boîte de CD ;
- 2 vues pour localiser la conserve de ravioli.

La fonction de visibilité est une fonction en escalier qui représente le nombre de vues dans lesquelles une portion du segment est incluse. L'acquisition des vues est interrompue dès que la totalité du segment d'intérêt a été examinée. Dans le cas de la séquence permettant de passer de la boîte de CD à la boîte de conserve, on peut remarquer que la seconde vue, permet non seulement de visualiser une nouvelle partie du segment mais également de décider entre les trois hypothèses en compétition dans la première vue. En effet, le changement de points de vue a permis de supprimer deux faux positifs.

Malgré des différences de points de vue entre la caméra déportée et la caméra embarquée, les objets sont localisés. En particulier pour la dernière séquence permettant d'aller de la boîte de CD à la boîte de conserve. La caméra embarquée donne une vue de dessus de la boîte de conserve, tandis que la vue déportée en donne une vue de côté. Très peu de points SIFT sont mis en correspondance sur la boîte elle-même. Un poids important est donné à ces points qui sont très proches de la ligne de vue. La localisation de l'objet est également possible grâce aux mises en correspondance de la texture avoisinant la boîte. Ce dernier exemple de localisation est un cas limite. Si le fond était uniforme, et si la vue de dessus, donnée par la caméra embarquée, ne permettait pas de voir de portion de l'étiquette de la boîte, la localisation échouerait vraisemblablement, par défaut de mise en correspondance. Pour élargir la méthode à des transformations affines, des descripteurs robuste à ces transformations pourraient être utilisés à la place des descripteurs SIFT, par exemple les descripteurs proposés par Mikolajczyk et Schmid [Mikolajczyk 04].

Un autre cas d'échec peut survenir si le motif avoisinant le *clac* correspond à un même motif présent à plusieurs emplacements sur la ligne de vue. La figure 4.20 illustre un cas limite. Dans cet exemple, plusieurs stylos feutres identiques sont positionnés parallèlement aux autres et de façon à être traversés par un même rayon issu du centre optique de la caméra déportée. Dans ce cas, des points sont mis en correspondance sur chaque stylo et la probabilité sur la position de l'objet est multimodale impossible de dissocier les hypothèses en compétition. Cet exemple semble très artificiel, cependant, sans que plusieurs objets identiques soient ainsi disposés, il peut arriver que plusieurs motifs le long du parcours de la ligne de vue donnent une réponse identique. Dans ce cas, une solution efficace pour résoudre le problème est simplement de proposer à l'utilisateur les deux positions, maxima de la vraisemblance, et de le laisser choisir la position qui correspond à l'objet qu'il souhaite saisir.

Conclusion

La méthode de mise en correspondance basée sur les points SIFT permet d'estimer grossièrement la position de l'objet. Notons que cette méthode de localisation fonctionne avec toute méthode de mise en correspondance dès lors qu'elle permet de détecter l'objet dans la vue de la caméra embarquée. Par exemple, des méthodes basées sur les histogrammes de couleur pourraient être utilisées à la place ou en complément des SIFT.

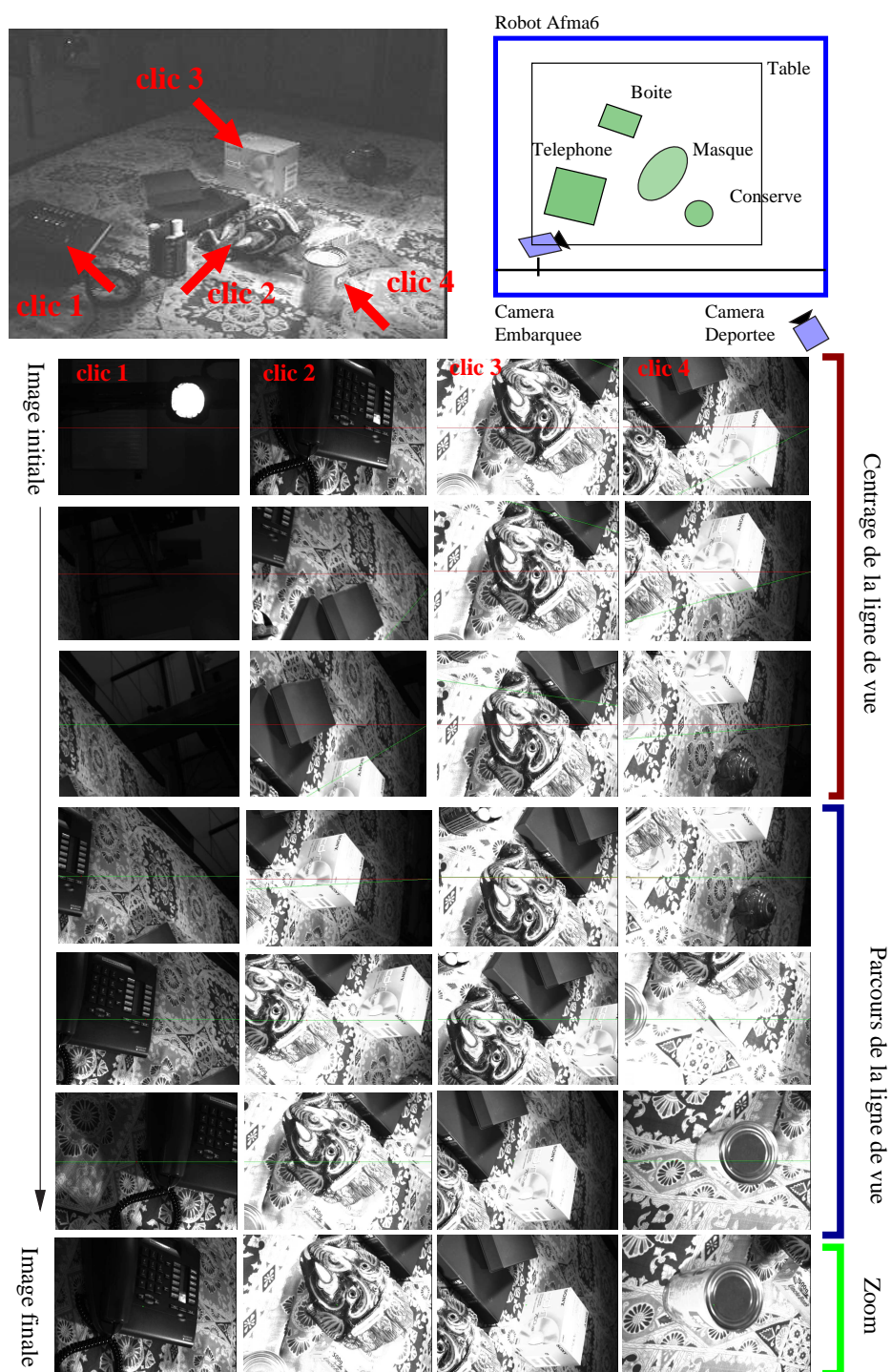


FIG. 4.18 – Plusieurs objets sont tour à tour sélectionnés par un clic, puis localisés et finalement centrés dans l'image de la caméra embarquée.

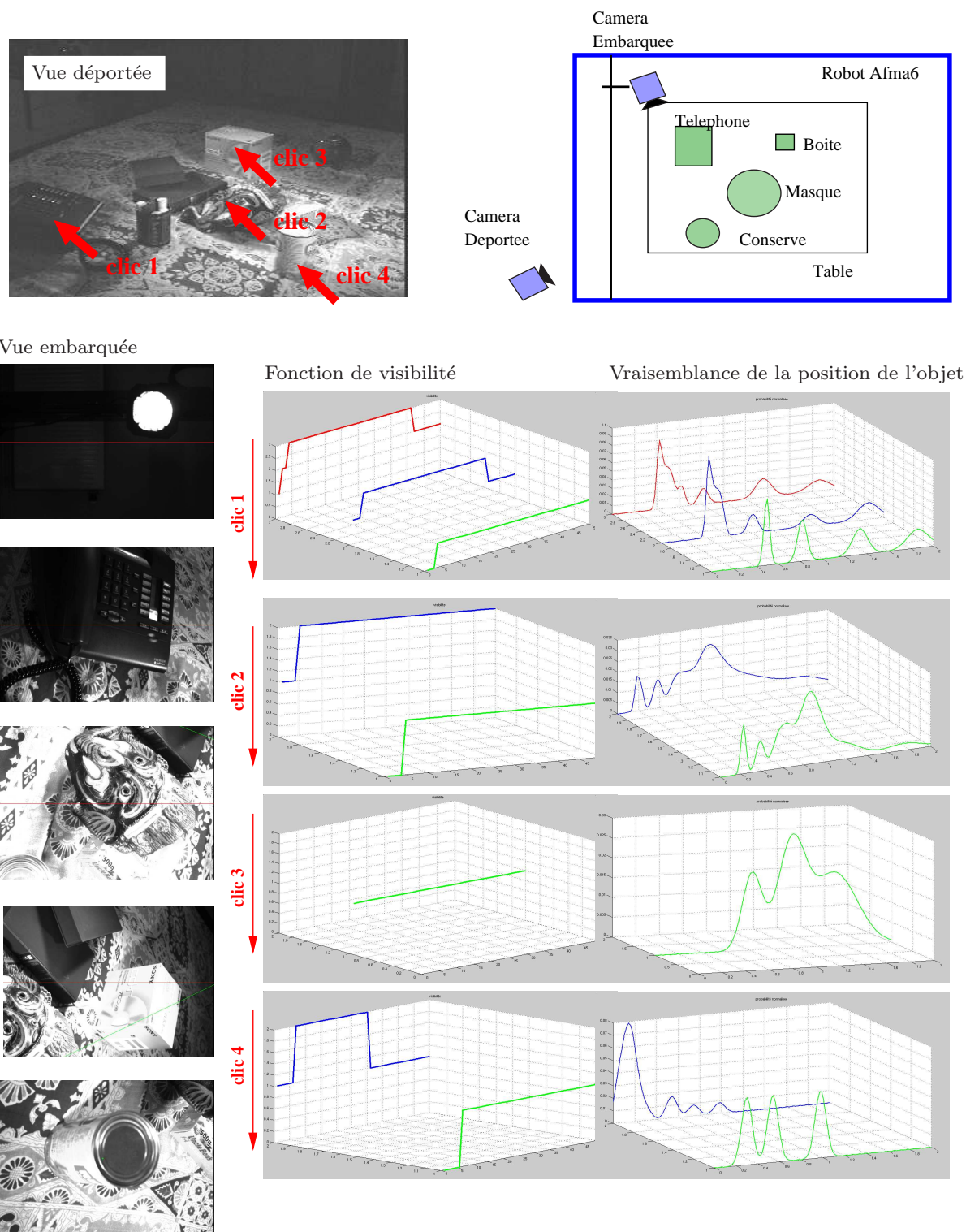
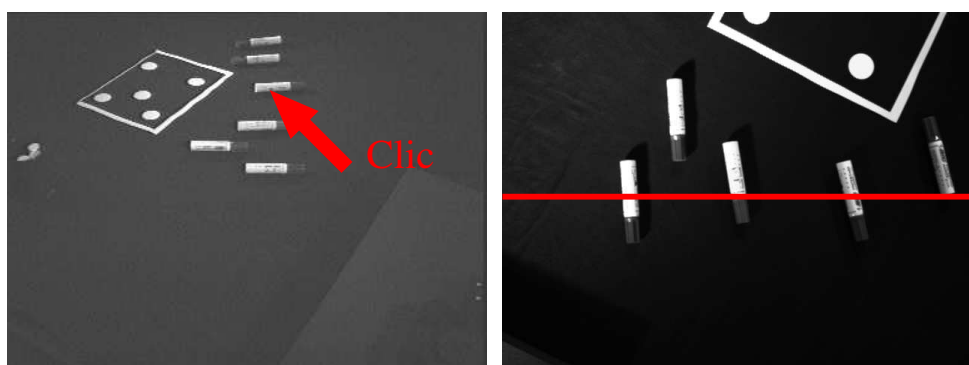


FIG. 4.19 – Plusieurs objets sont tour à tour électionnés, localisés et centrés dans l'image de la caméra embarquée. La colonne de gauche présente les images acquises par la caméra embarquée au moment de la sélection de l'objet suivant dans la vue déportée. La seconde colonne présente les fonction de visibilité et la troisième colonne présente l'estimation de la vraisemblance de la position de l'objet. Les courbes vertes correspondent à l'évaluation des fonctions après l'acquisition d'une première vue, les bleues après deux vues et les rouges après trois vues.



Caméra déportée : clic

Caméra embarquée : parcours de la ligne

FIG. 4.20 – Cas limite de l’algorithme de localisation : plusieurs objets semblable à l’objet d’intérêt se trouve sur la ligne de vue.

La mise en commun des informations contenues dans la séquence d’images se fait très simplement et de manière itérative en fusionnant l’estimation de la position de l’objet dans la vue courante à celle obtenue dans les vues précédentes. Le problème du recouvrement des vues successives est résolu en utilisant une fonction de visibilité qui permet de traiter les informations redondantes, c’est-à-dire les points détectés dans plusieurs vues qui se chevauchent. Ainsi, les vues sont prises en compte à intervalle de temps régulier sans se soucier de la redondance potentielle de l’information qu’elles contiennent.

Les points SIFT peuvent être détectés à chaque pas du processus dans les deux images ou bien uniquement dans l’image de la caméra embarquée. L’avantage de la détection simultanée dans les deux images est qu’elle est robuste aux changements globaux d’illumination puisque les points sont détectés au même instant. Pour gagner en temps de calcul, les points SIFT détectés dans les deux images pourraient être suivi en utilisant une méthode KLT, par exemple. La détection des points d’intérêt n’est alors nécessaire que lorsqu’un trop grand nombre de points sont perdus par le suivi. L’inconvénient de la détection par des points SIFT dans la vue déportée à chaque pas du processus est quelle nécessite une prise en compte des occultations qui peuvent advenir lorsque le bras passe entre la caméra déportée et l’objet.

La localisation obtenue permet de centrer l’objet dans l’image de la caméra déportée en fin d’asservissement. Ainsi l’objet est dans le champ de vision des deux caméras à la fin du processus. Cette méthode peut, par exemple, être utilisée pour remplacer la phase de positionnement manuelle de la caméra embarquée du scénario AVISO. Cependant, l’objet n’est pas segmenté sur le fond. La zone localisée est le voisinage du *clic*. Cette méthode ne donne aucune information sur la forme de l’objet ni sur ses dimensions. La localisation de l’objet permet à la pince d’approcher grossièrement, reste ensuite à déterminer quelle doit être son orientation et la position de ses mors pour saisir l’objet. Pour adapter la saisie à la forme et aux dimensions de l’objet, il faut donc parvenir à estimer ces caractéristiques à l’aide du système de perception du robot. La seconde partie de ce manuscrit est consacrée à la description plus précise de la forme de l’objet.

D’autre part, une fois l’objet grossièrement localisé, il est possible d’acquérir une vue où il apparaît centré et ainsi d’avoir une image plus détaillée de l’objet que celle donnée par la vue déportée. Cette image de meilleure qualité peut être particulièrement intéressante pour utiliser les algorithmes classiques de reconnaissance d’objet qui auraient pu échouer en utilisant l’image de la caméra déportée. L’estimation de la position de l’objet permet de déplacer la caméra embarquée sur une sphère de vision centrée sur l’objet afin d’acquérir les vues permettant de caractériser sa forme et le saisir.

Dans cette première partie, deux contributions de cette thèse ont été présentées : un asservissement

permettant de parcourir la ligne de vue issue du *clic* et une méthode de localisation d'un objet inconnu basée sur la mise en correspondance de descripteurs SIFT dans la vue déportée et dans la vue embarquée. Dans la seconde partie de cette thèse deux autres contributions seront présentées :

- **L'estimation en ligne de la forme approximative d'un objet grossièrement convexe**, c'est-à-dire son orientation et ses dimensions ;
- **La sélection active des meilleurs vues** pour affiner l'estimation des paramètres décrivant la forme de l'objet.

Deuxième partie

Caractérisation de la forme d'un objet inconnu et exploration par vision active

Le chapitre précédent a présenté une méthode permettant de localiser un *objet générique* à partir d'un *clic* sur l'image acquise par une caméra déportée. Cependant, si aucune hypothèse n'est formulée *a priori* sur la forme et l'orientation spatiale de l'objet, la seule donnée de sa position n'est pas suffisante pour adapter l'approche de la pince et il est alors nécessaire d'obtenir d'autres informations. Pour respecter les critères énoncés dans le premier chapitre de cette thèse (voir la section 1.1.3), la méthode développée doit permettre de saisir des *objets génériques* (hypothèse 3) positionnés de manière quelconque, dans une scène statique (hypothèse 2)². D'autre part, pour être employée sur un système réel, en milieu humain, l'assistance technique à la saisie que nous proposons doit être robuste aux bruits de mesure et aux imprécisions de la calibration. Enfin, pour être acceptée par l'utilisateur, elle doit permettre, au pire, la saisie en un temps comparable à une saisie en mode téléopéré.

Dans cette thèse, nous nous plaçons dans le cadre d'une approche multi-vues. La littérature traitant de la reconstruction 3D d'objets et de scènes à partir d'informations visuelles est très fournie. Des résultats remarquables ont été obtenus dans le domaine de la reconstruction 3D volumique de scènes statiques ou dynamiques. Par exemple, en détectant la silhouette d'un objet placé sur un fond uniforme (*Visual hull*), par exemple [Franco 06], ou en supposant la scène Lambertienne et en reconstruisant l'objet sous la forme d'un volume discret dont les éléments sont creusés ou conservés selon leur *photo-consistance* [Steven 97, Kutulakos 99]. Même si des tentatives ont été faites pour proposer des versions temps-réel de ces méthodes, par exemple, en proposant des approches pyramidales pour les reconstructions voxelique [Prock 98] ou une parallélisation des calculs pour les méthodes basées silhouette, les méthodes de reconstruction volumique sont assez lourdes à manipuler et gourmandes en temps de calcul. En ce sens, ces méthodes sont bien adaptées pour des applications où les calculs peuvent être faits hors ligne. Le niveau de détail obtenu avec une résolution voxelique importante permet d'envisager la création de personnages pour les jeux vidéos ou encore la reconstruction 3D de sculptures pour sauvegarder une trace du patrimoine conservé dans les musées. Il est cependant légitime de se demander si une reconstruction 3D aussi précise de l'objet est nécessaire à sa saisie, surtout lorsque l'outil de préhension est une pince simple à deux doigts.

Pour saisir un *objet générique* à partir d'informations calculées en ligne, une première stratégie de saisie consiste à utiliser une reconstruction précise de l'objet et calculer ensuite les meilleurs points de saisie physique [Saut 07, Michel 06]. Une autre voie d'étude tente de mimer les prises qu'auraient adopté un humain, par exemple, en repérant sur l'objet l'emplacement de poignées, manches et anses [Edsinger 06, ElKhoury 07]. D'autres approches consistent à approximer la forme de l'objet par un ensemble de primitives géométriques et à calculer les orientations de saisie possibles sur cette forme simplifiée [Huebner 08a, Speth 08]. La motivation de cette thèse est de proposer une méthode de saisie reposant sur le calcul de primitives simples. L'idée sous-jacente est qu'il n'est pas nécessaire de recourir à des méthodes de reconstruction 3D fines pour parvenir à la saisie de la plupart des objets.

De la modélisation de l'objet retenue pour la saisie dépend la méthode d'acquisition des données : pour obtenir une reconstruction exhaustive d'un objet, le système de perception doit être étalonné et considéré comme peu bruité. À l'inverse, si seule la forme générale de l'objet est recherchée, quelques erreurs d'étalonnage peuvent être tolérées. Dans le cadre de l'utilisation d'un système de vision, la stratégie de saisie a également un impact sur le choix des primitives à extraire dans les images : points d'intérêt, contours, régions d'une même couleur.

L'utilisation d'un système de vision pour trouver les informations nécessaires à la saisie induit le problème du positionnement optimal des caméras. En effet, si les positions sont bien choisies, alors les caractéristiques de l'objet sont calculées sans ambiguïtés. À l'inverse, il est possible que les informations calculées soient erronées. En particulier lorsque deux vues sont proches, l'information sur la dimension en profondeur de l'objet est évidemment mal définie. Dans ce cas, l'idéal est d'ajouter une autre vue perpendiculaire aux premières pour ajouter un maximum d'information sur les caractéristiques de l'objet.

²L'ensemble des hypothèses est rappelé en annexe C

Trois niveaux peuvent être distingués : au plus haut niveau, la stratégie de saisie et le choix de modélisation, puis la perception qui permet la modélisation et enfin, l'extraction des informations dans plusieurs vues dont la position peut être judicieusement choisie.

Dans le premier chapitre de cette partie, la stratégie globale de saisie choisie dans le cadre de notre application sera présentée. Il s'agit d'approximer la forme des objets par des fonctions quadratiques. Le second chapitre rappelle les propriétés de base de ces fonctions paramétriques et le troisième chapitre présente une méthode de reconstruction d'une quadrique à partir de ses projections dans un ensemble de vues. Nous montrerons aussi que le choix des vues influence la reconstruction. En conséquence, le chapitre suivante présentera une méthode de *vision active* permettant de sélectionner les meilleures vues selon un critère relatif à l'information qu'elles contiennent. Finalement des résultats de reconstructions avec un choix des vues par vision active seront présentés. Enfin, au plus bas niveau, le dernier chapitre présente une méthode permettant d'extraire les contours des objets dans des images réelles et les résultats de reconstruction obtenus pour des objets réels.

Types d'objets et stratégie de saisie

5.1 Hypothèses générales

Rappelons que l'objectif pratique soutenant cette étude est d'étendre les capacités de saisie du système Aviso, présenté dans le premier chapitre, à la saisie d'*objets génériques* se trouvant dans une position quelconque. Parmi ces objets, on peut distinguer les objets rigides (tasse, crayon, boîtes, etc.) de ceux qui ne le sont pas (jeux de clés, grappe de raisin, tissu, sachet, etc...). Comme il l'a déjà été spécifié dans le premier chapitre, cette étude traite le cas des objets rigides (hypothèse 4)¹.

En observant les objets rigides du quotidien, des structures communes qui sont destinées à simplifier leur utilisation peuvent être identifiées : les conditionnements alimentaires sont généralement des contenants parallélépipédiques ou cylindriques (voir la figure 5.1) et les objets manufacturés présentent des axes de symétrie, ou axes naturels de préhension [Michel 06] voire des poignées ou des anses (voir figure 5.2).

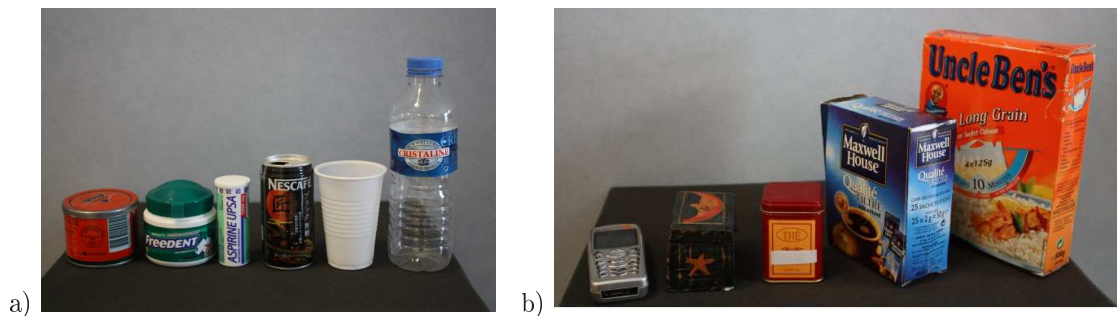


FIG. 5.1 – Objets rigides, globalement convexes et présentant des symétries : a) objets cylindriques (bouteilles, cannettes, conserves, etc.) b) objets parallélépipédiques (boîtes, téléphone, etc.)

Selon le volume des mors de la pince considérée, saisir une tasse par son anse n'est pas toujours possible (voir figure 5.2). D'autre part, ce n'est pas forcément une prise très stable en comparaison aux surfaces latérales de la tasse. Ainsi, une anse n'est pas nécessairement une prise et peut même être considérée comme un obstacle à la saisie. Dans le cadre de cette étude, la saisie sera limitée aux objets

¹L'ensemble des hypothèses est rappelé en annexe C

rigides qui ne présentent ni anse, ni poignée, ni protubérances. Nous parlerons d'objets *globalement convexes*. À première vue, cette hypothèse semble forte, cependant, elle est vérifiée par un grand nombre d'objets du quotidien (voir figure 5.1).



FIG. 5.2 – Deux objets présentant des prises adaptées à une saisie humaine. Ces prises ne sont pas nécessairement les mieux adaptées à une saisie robotique, en particulier lorsque la pince possède des mors trop volumineux pour être glissés dans l'anse.

Hypothèse 9 *L'objet est de forme globalement convexe.*

Pour faciliter leur manipulation, les objets conditionnés ou manufacturés présentent des axes de symétrie. Cette hypothèse a déjà été exploitée avec succès pour segmenter un grand nombre d'objets sur un arrière plan quelconque [Li 08] et pour déterminer les axes de saisie d'objets plans [Blake 95, Sanz 05] ou 3D [Hauck 99].

Hypothèse 10 *L'objet présente un axe de symétrie.*

Comme dans la plupart des travaux de saisie utilisant des capteurs extéroceptifs, les objets sont considérés de densité massique uniforme. D'autre part, comme les objets sont considérés convexes, l'intérieur des contenants sera considéré plein.

Hypothèse 11 *La densité massique de l'objet est uniforme.*

5.2 La stratégie de saisie

En supposant que l'objet à saisir est rigide, globalement convexe, qu'il présente des symétries et est de densité massique uniforme, alors la stratégie de saisie consiste à approcher la pince perpendiculairement à l'axe principal de l'objet en l'alignant avec son plus petit axe. La figure 5.3 illustre cette stratégie de saisie pour quatre objets. Les deux premiers objets respectent les hypothèses fixées et les deux autres présentent des non convexités.

Néanmoins, tant que les excroissances sont suffisamment petites ou placées de façon à ne pas obstruer le chemin de la pince, cette stratégie reste valide. Dans le cas contraire, rien ne garantit son succès et il est nécessaire de détecter ces obstacles et d'en tenir compte pour le positionnement de la pince. La figure 5.4 présente quelques objets non convexes. Dans l'image de gauche, ces trois objets se trouvent dans une pose favorable à notre stratégie de saisie. Aucune convexité non détectable ne fait obstacle à l'approche de la pince. Dans l'image de droite, ces trois objets sont dans une position moins favorable. En particulier, les pattes des animaux-jouets risquent de perturber l'approche de la pince.

Il peut d'ores et déjà être remarqué que cette stratégie engendre plusieurs solutions d'approche de la pince. En effet, la boîte peut être saisie par la droite ou par la gauche, la canette qui est cylindrique peut être saisie depuis n'importe quelle direction, du moment que la pince est perpendiculaire au

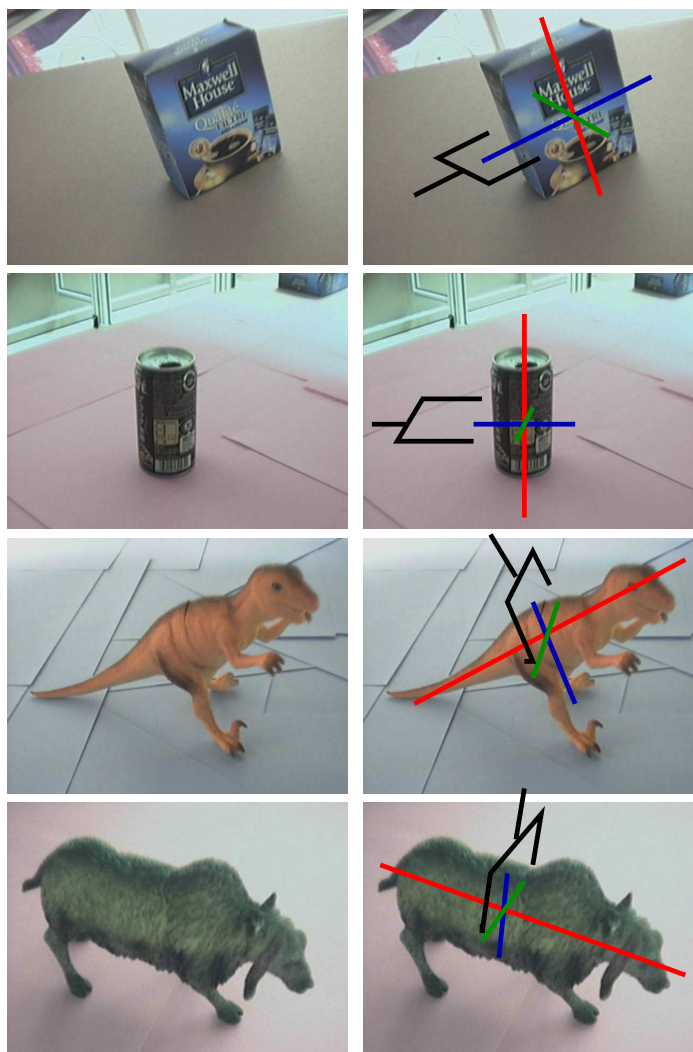


FIG. 5.3 – Stratégie de saisie choisie pour des objets génériques et globalement convexes. La pince est alignée avec les axes principaux de l'objet et centrée sur son centre de gravité. En rouge l'axe principal de la pince, en bleu l'axe moyen et en vert le petit axe.



FIG. 5.4 – Si les objets sont globalement convexes et présentent uniquement quelques concavités suffisamment petites devant les dimensions de la pince, alors ces concavités ne perturbent pas la saisie. À gauche, les objets sont dans une pose favorable à la saisie, à droite, les pattes des jouets risquent de gêner l'approche de la pince.

grand axe. Par ailleurs, dans le cas des deux jouets, parmi les deux solutions de saisie (depuis le haut ou depuis le bas), une seule solution est acceptable puisque l'autre conduirait la pince à heurter la table. Pour choisir l'approche la plus adaptée, il suffit de faire appel à quelques heuristiques simples : en supposant qu'un objet est généralement posé sur un support, seule une saisie "depuis le haut" est possible, si les deux options de saisie respectent cette règle alors la solution choisie sera celle qui demande le moins de déplacement. Si le système dispose d'un module de détection d'obstacle, cette deuxième règle peut être remplacée par : la solution choisie est celle qui permet d'éviter les obstacles.

5.3 Modélisation retenue

Les informations nécessaires à la saisie d'un objet sont donc les directions de ses axes principaux et la position de son centre de gravité. Pour obtenir ces informations, la forme de l'objet peut être approximée par une fonction paramétrée de type quadratique. Cette représentation présente quelques propriétés intéressantes pour notre application (voir le chapitre 6) :

- **Une représentation compacte** : une quadrique est représentée par 9 paramètres ;
- **Informations pour la saisie** : les 9 paramètres décrivant une quadrique contiennent l'information de sa position, de son orientation, de sa forme et de ses dimensions, c'est-à-dire directement toutes les informations nécessaires à la saisie ;
- **Projection** : la projection perspective d'une quadrique est une conique. Nous montrerons qu'en mesurant un ensemble de coniques dans un ensemble de vues de l'objet, il est possible de remonter aux paramètres de la quadrique. Ainsi, l'estimation des paramètres de la quadrique correspondant au mieux à la forme de l'objet demande l'extraction des contours de l'objet dans un ensemble d'images.

Notons que ces caractéristiques ont été utilisés par Flandin *et al.* pour décrire les objet composant une scène [Flandin 02]. Ils traduisent les données du centre de l'objet, de son orientation et de ses dimensions en des variable aléatoires qui seront estimées au cours du processus de reconstruction. Le cadre probabiliste leur permet d'inclure les incertitudes de mesure directement dans la modélisation de l'objet et d'obtenir une représentation qui peut être aisément mise à jours à chaque ajout d'information. D'autre part, la représentation choisi permet de sélectionner les meilleurs vues par vision active. Un autre avantage de cette méthode est qu'elle permet d'évaluer les caractéristiques de plusieurs objets simultanément.

5.4 Acquisition des données

La forme de l'objet peut donc être approximée par une fonction quadratique dont les paramètres sont estimés à partir de la détection des contours dans un ensemble d'images. Il est donc nécessaire de disposer d'un ensemble de vues de l'objet. En utilisant les résultats du chapitre précédent, nous supposerons qu'une estimation grossière de la position de l'objet est disponible. Elle est obtenue à partir de la seule information donnée par les coordonnées du *clac* (voir le chapitre 4).

Hypothèse 12 *Une estimation grossière de la position de l'objet est donnée.*

Les vues de l'objet sont alors acquises sur une sphère de vision centrée sur la position estimée de l'objet. Nous verrons dans le chapitre 9 comment sélectionner ces vues pour améliorer la reconstruction.

Pour extraire les contours des objets dans ces vues, nous supposerons que ceux ci peuvent être segmentés sur le fond à partir de l'information donnée par le *clac* dans la vue globale de la scène. Cette étape de segmentation n'est pas triviale et les systèmes de saisie qui y ont recours imposent des hypothèses fortes sur l'objet et la scène (voir 8.1). Dans le cadre de cette étude, un point de

l'objet est connu dans la vue de la caméra déportée et l'information donnée par ce point permet de connaître grossièrement la position de l'objet. Nous montrerons dans la section 8.1 comment exploiter ces informations pour détecter le contour de l'objet.

Modélisation d'un objet par une quadrique

La stratégie de saisie choisie consiste à approcher la pince qui équipe le système robotique perpendiculairement à l'axe principal de l'objet. C'est une stratégie de saisie très simple qui est bien adaptée aux objets présentant de fortes symétries, globalement convexes et ont une répartition massique homogène (voir les hypothèses en annexe C). Pour parvenir à une telle saisie, les informations qui caractérisent l'objet d'intérêt sont :

- ses axes principaux ;
- son centre d'inertie ;
- ses dimensions principales.

Dans ce chapitre, nous montrerons tout d'abord qu'approximer la forme d'un objet par une quadrique permet d'obtenir une modélisation qui contient toutes ces informations. Le second paragraphe présentera les propriétés projectives des quadriques sur lesquelles se basent les méthodes de reconstruction qui sont présentées dans le chapitre suivant.

6.1 Définitions

L'objet à saisir est inclus dans une enveloppe paramétrable par une fonction implicite $h(\mathbf{X}, \mathcal{P})$, où \mathbf{X} est un point appartenant à la surface de la fonction et \mathcal{P} représente l'ensemble de paramètres de la fonction. Cet objet est observé depuis N points de vue par une caméra et ses contours dans les N plans image sont inclus dans des fonctions génératrices $g(\mathbf{x}, \rho)$, où \mathbf{x} est un point du plan image et ρ les paramètres 2D de la génératrice.

En supposant que l'objet à saisir est rigide, de densité massique homogène et globalement convexe (d'après les hypothèses 4, 11 et 9), cette fonction paramétrée peut avoir une forme quadratique. La famille des quadriques d'un espace affine de dimension n est assez vaste puisqu'elle regroupe tous les polynômes de degrés 2 à n variables. Notons qu'en dimension 2, une quadrique est appelée une *conique*. Dans la suite de ce chapitre, nous parlerons indifféremment de quadriques ou de fonctions quadratiques.

L'expression cartésienne d'une quadrique en dimension 3 s'écrit :

$$h(\mathbf{X}, \mathcal{P}) = Q(\mathbf{X}) = q_1 X^2 + q_2 Y^2 + q_3 Z^2 + 2q_4 XY + 2q_5 XZ + 2q_6 YZ + 2q_7 X + 2q_8 Y + 2q_9 Z + q_{10} \quad (6.1)$$

où tout point de coordonnées homogènes satisfaisant l'équation $\mathcal{Q}(\mathbf{X}) = 0$ appartient à la surface de la quadrique.

En utilisant les propriétés de l'algèbre bilinéaire, il est possible de classer les quadriques en trois types. En effet, il existe une base dans laquelle \mathcal{Q} est nécessairement de l'un de ces trois types, [Ber90] :

- type I : $\mathcal{Q} = q_1X^2 + q_2Y^2 + q_3Z^2$;
- type II : $\mathcal{Q} = q_1X^2 + q_2Y^2 + q_3Z^2 + 1$ et $q_1q_2q_3 \neq 0$;
- type III : $\mathcal{Q} = q_1X^2 + q_2Y^2 - 2Z$.

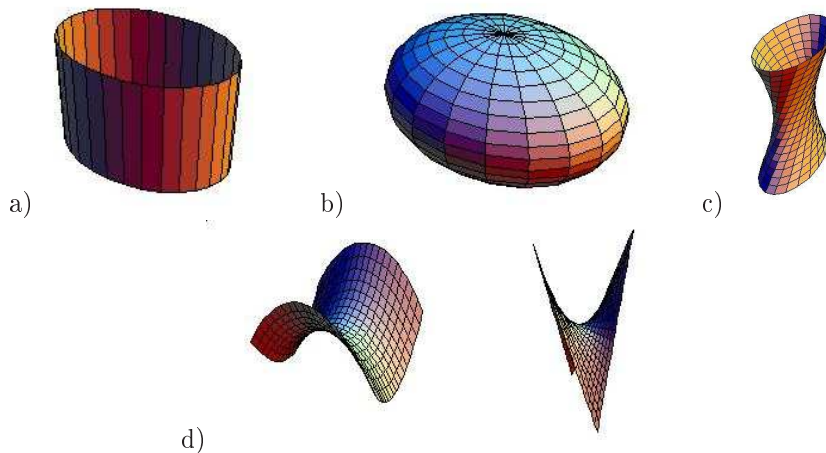


FIG. 6.1 – Quelques exemple de quadriques de dimension 3 : a) cylindre elliptique, b) ellipsoïde, c) paraboloides elliptiques, d) hyperboloïdes.

Le type I contient les points et les droites ; le type II, les cylindres, les ellipsoïdes, les hyperboloïdes à une nappe et à deux nappes ; le type III contient les paraboloides elliptiques et les hyperboloïdes elliptiques. Notons que, parmi ces formes, seules les ellipsoïdes ont un volume fini. Quelques exemples de quadriques tridimensionnelles sont donnés par la figure 6.1.

Les primitives quadratiques offrent un modèle de description des surfaces courbes d'une grande généralité allant même jusqu'à décrire des surfaces courbes à points complexes qui n'ont aucun sens dans le cadre de notre application. D'autre part, la représentation \mathcal{Q} n'est pas minimale, ce qui peut faire échouer les méthodes d'estimation des paramètres que nous présenterons par la suite. Dans le reste de ce chapitre, nous montrerons comment poser le problème de l'estimation des paramètres pour résoudre ces deux points.

Dans l'espace projectif de dimension 3, \mathbb{P}^3 , une quadrique est une surface définie sous une forme matricielle par l'équation :

$$\mathcal{Q} = \mathbf{X}^\top \mathbf{\Gamma} \mathbf{X} \quad (6.2)$$

où $\mathbf{\Gamma}$ est la matrice symétrique 4×4 associée à la quadrique \mathcal{Q} :

$$\mathbf{\Gamma} = \begin{pmatrix} q_1 & q_4 & q_5 & q_7 \\ q_4 & q_2 & q_6 & q_8 \\ q_5 & q_6 & q_3 & q_9 \\ q_7 & q_8 & q_9 & q_{10} \end{pmatrix} \quad (6.3)$$

et une matrice $\mathbf{\Gamma}_{33}$ peut être définie et associée à quadrique \mathcal{Q} :

$$\mathbf{\Gamma}_{33} = \begin{pmatrix} q_1 & q_4 & q_5 \\ q_4 & q_2 & q_6 \\ q_5 & q_6 & q_3 \end{pmatrix} \quad (6.4)$$

Remarque : L'étude des caractéristiques des matrices $\mathbf{\Gamma}$ et $\mathbf{\Gamma}_{33}$ (rangs $R_{\mathbf{\Gamma}}$ et $R_{\mathbf{\Gamma}_{33}}$, signe du déterminant $\Delta_{\mathbf{\Gamma}}$) permet également d'obtenir la classification des fonctions quadratiques en trois types.

La représentation matricielle contient donc toutes les informations permettant de définir la quadrique et il est aisé de passer d'une représentation à l'autre. De plus la représentation matricielle permet des simplifications d'écriture, notamment en ce qui concerne le changement de repère d'une quadrique. Dans l'espace projectif \mathbb{P}^3 , le changement de repère d'expression d'une quadrique est effectué de manière similaire au changement de repère d'un point ${}^a\mathbf{X}$ d'un repère \mathcal{R}_a à un repère \mathcal{R}_b définie par l'équation (2.3).

Soit ${}^b\mathbf{M}_a$, la transformation homogène à appliquer à la quadrique \mathcal{Q} , le changement de repère d'expression impose la transformation suivante sur la matrice associée $\mathbf{\Gamma}$:

$${}^b\mathbf{\Gamma} = {}^b\mathbf{M}_a^{-\top} {}^a\mathbf{\Gamma} {}^b\mathbf{M}_a^{-1} \quad (6.5)$$

$${}^b\mathbf{\Gamma} = {}^a\mathbf{M}_b^{\top} {}^a\mathbf{\Gamma} {}^a\mathbf{M}_b \quad (6.6)$$

6.2 Calcul des informations nécessaires à la saisie

Les caractéristiques de l'objet peuvent être assimilées à celles de la quadrique qui correspond au mieux à sa forme. Le centre de la quadrique, la direction de ses axes principaux et ses dimensions le long de ses axes peuvent être obtenus à partir de l'estimation de ses coefficients. Ces informations serviront à orienter la pince pour adapter l'approche à la pose et aux dimensions de l'objet.

6.2.1 Centre d'une quadrique

Dans une base orthonormée propre, les termes croisés XY , XZ et YZ peuvent être annulés en effectuant les rotations adéquates et l'équation (6.1) s'écrit alors :

$$\mathcal{Q}(\mathbf{X}) = q_1X^2 + q_2Y^2 + q_3Z^2 + 2q_4X + 2q_5Y + 2q_6Z + q_7$$

En réordonnant, l'équation ci dessus devient :

$$\mathcal{Q}(\mathbf{X}) = q_1\left(X + \frac{q_4}{q_1}\right)^2 + q_2\left(Y + \frac{q_5}{q_2}\right)^2 + q_3\left(Z + \frac{q_6}{q_3}\right)^2 + q_{10}$$

Une autre manière de calculer le centre de la quadrique est de chercher les coordonnées du point $\bar{\mathbf{X}}_c$ pour lequel les dérivées partielles $\frac{\partial \mathcal{Q}}{\partial \bar{\mathbf{X}}}$ s'annulent. Cela permet de calculer les coordonnées du centre sans changer de base. Le centre de l'objet peut ainsi être estimé en résolvant le système d'équations suivant :

$$\mathbf{\Gamma}_{33}\bar{\mathbf{X}}_c = \mathbf{b}$$

avec $\mathbf{b} = (-q_7 \quad -q_8 \quad -q_9)^\top$ et $\bar{\mathbf{X}}_c = (X_c \quad Y_c \quad Z_c)^\top$.

6.2.2 Axes principaux et dimensions

Les propriétés géométriques d'une quadrique \mathcal{Q} se reflètent dans les propriétés algébriques de la matrice symétrique réelle $\mathbf{\Gamma}_{33}$ qui lui est associée.

Soit \mathbf{v}_i et λ_i les vecteurs propres et les valeurs propres de $\mathbf{\Gamma}_{33}$, alors les axes principaux de \mathcal{Q} correspondent aux directions décrites par les vecteurs \mathbf{v}_i .

Le calcul des trois vecteurs propres permet de définir l'orientation de la quadrique dans un repère de base \mathcal{R}_b . En effet, la matrice $\mathbf{\Gamma}_{33}$ est une matrice symétrique réelle, elle est donc diagonalisable dans une base propre définie par les trois vecteurs propres \mathbf{v}_i . Il suffit de normaliser ces trois vecteurs pour obtenir une base orthonormée, $|\mathbf{v}_i| = 1$. Le calcul du déterminant de la matrice $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ indique si cette base est directe ($\det(\mathbf{V}) = 1$) ou indirecte ($\det(\mathbf{V}) = -1$).

Pour construire la matrice de rotation, notons \mathbf{v}_1 le vecteur propre associé au plus grand axe ie à la valeur propre la plus grande, \mathbf{v}_2 le vecteur propre associé à l'axe moyen, et \mathbf{v}_3 le vecteur propre donnant la direction du plus petit axe. Pour garantir que la base orthonormée définie par ces trois vecteurs est directe, on construit \mathbf{v}_3 tel que $\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2$.

$${}^b\mathbf{R}_q = (\mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3)$$

Les valeurs propres de $\mathbf{\Gamma}_{33}$ nous renseignent sur les dimensions relatives des demi axes de la quadrique \mathcal{Q} . Pour calculer leurs valeurs métriques, il faut calculer les coefficients de la quadrique dans sa base propre, normaliser l'équation telle que $k_{10=1}$ et calculer les valeurs propres dans la base propre. Ainsi les mesures des demi grands axes valent $d_i = \frac{1}{\lambda_i^2}$.

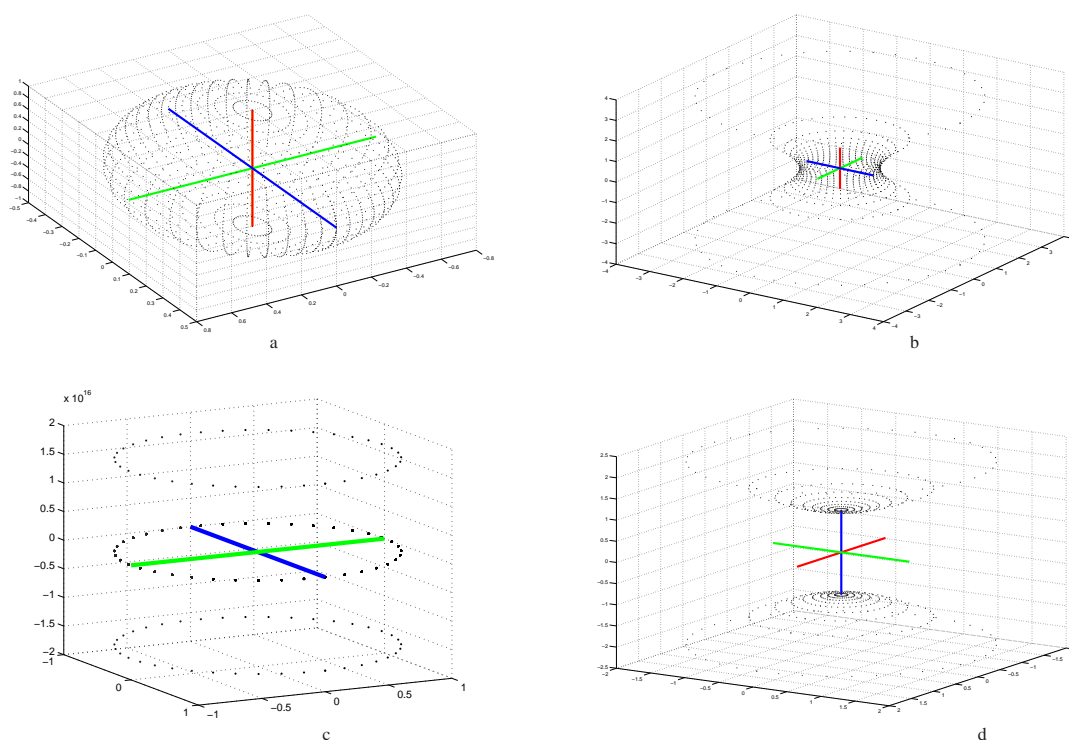


FIG. 6.2 – Axes principaux d'une quadrique par ordre décroissant de taille : rouge, vert puis bleu. a) ellipsoïde $\mathcal{Q} = (4, 2, 1, 0, 0, 0, 0, 0, -1)$ b) hyperboloïde à une nappe $\mathcal{Q} = (1, 1, -1, 0, 0, 0, 0, 0, -1)$ c) cylindre $\mathcal{Q} = (1, 1, 0, 0, 0, 0, 0, 0, -1)$ d) hyperboloïde à deux nappes $\mathcal{Q} = (-1, -1, 1, 0, 0, 0, 0, 0, -1)$

L'estimation des paramètres de la quadrique approximant la forme d'un objet permet d'obtenir directement les caractéristiques nécessaires à la saisie. Le paragraphe suivant présente d'autres propriétés intéressantes des quadriques, que sont leurs propriétés projectives. Elles représentent le lien entre une quadrique et sa projection dans des plans images, ce qui nous permettra ensuite d'établir un schéma d'estimation des paramètres d'une quadrique à partir d'informations vidéo.

6.3 Propriétés projectives des quadriques

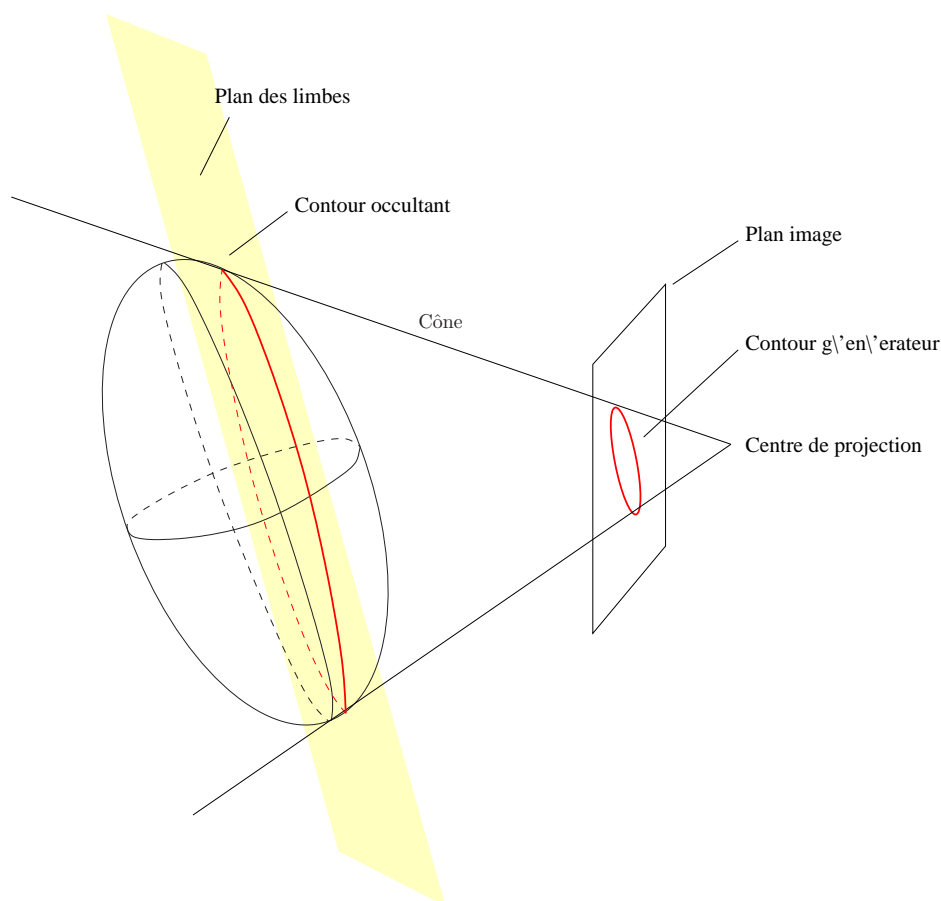


FIG. 6.3 – Projection perspective d'un ellipsoïde.

Ce paragraphe présente les propriétés projectives d'une quadrique, c'est-à-dire les positions des points obtenus par projections de l'ensemble des points de sa surface sur le plan image d'une caméra. Par construction, toutes les droites qui passent par le centre de projection et le contour occultant d'une quadrique sont tangentes à la quadrique (voir l'exemple illustratif figure 6.3). Il peut être démontré que le contour occultant d'une quadrique est une conique [Ma 94] et donc que la projection perspective d'une fonction quadratique Q à trois dimensions est une *conique* \mathcal{C} , c'est-à-dire une fonction quadratique de dimension 2 à 5 paramètres. La projection de la fonction implicite h dans un plan image est le *contour générateur*, noté g .

L'expression cartésienne d'une quadrique de dimension 2 s'écrit :

$$g(\mathbf{x}, \rho) = \mathcal{C}(\mathbf{x}) = k_1x^2 + k_2y^2 + 2k_3xy + 2k_4x + 2k_5y + k_6 \quad (6.7)$$

De la même manière que pour l'équation d'une quadrique de dimension 3, l'expression d'une *quadrique de dimension 2*, ou *conique* \mathcal{C} , peut s'écrire sous la forme matricielle suivante :

$$\mathcal{C} = \mathbf{x}^\top \mathbf{C} \mathbf{x} \quad (6.8)$$

où \mathbf{C} est la matrice symétrique de dimension 3×3 :

$$\mathbf{C} = \begin{pmatrix} k_1 & k_3 & k_4 \\ k_3 & k_2 & k_5 \\ k_4 & k_5 & k_6 \end{pmatrix} \quad (6.9)$$

La relation entre une quadrique et les coniques résultant de sa projection peut être exprimée sous une forme non linéaire dans l'espace projectif, ou sous une forme linéaire dans l'espace dual.

6.3.1 Dans l'espace projectif.

Ce paragraphe donne l'expression, dans l'espace projectif \mathbb{P}^3 , des coefficients d'une conique projetée en fonction des coefficients de la quadrique et de la position de la caméra.

Rappelons l'expression d'une quadrique de dimension 3, (6.1) :

$$\mathcal{Q}(\mathbf{X}) = q_1 X^2 + q_2 Y^2 + q_3 Z^2 + 2q_4 XY + 2q_5 XZ + 2q_6 YZ + 2q_7 X + 2q_8 Y + 2q_9 Z + q_{10}$$

D'après l'équation de projection perspective d'un point (2.6), $X = Zx$ et $Y = Zy$. En introduisant ces expressions de X et Y dans l'équation (6.1) et en divisant les termes par Z^2 , avec $Z \neq 0$, une équation du second degré en $\frac{1}{Z}$ est obtenue :

$$\mathcal{Q}(\mathbf{X}) = 0 \Leftrightarrow A \frac{1}{Z^2} + 2B \frac{1}{Z} + C = 0$$

avec

$$\begin{aligned} A &= q_{10} \\ B &= q_7 x + q_8 y + q_9 \\ C &= q_1 x^2 + q_2 y^2 + q_3 + 2q_4 xy + 2q_5 x + 2q_6 y \end{aligned}$$

Cette équation représente l'intersection de la ligne de vue passant par le point \mathbf{x} avec la quadrique au point \mathbf{X} . Or les points mesurés sur le plan image sont sur la conique résultant de la projection de la quadrique. La ligne de vue passant par ces points est tangente à la quadrique. Le point d'intersection entre cette ligne et la quadrique est donc unique. On en déduit donc :

$$\Delta = B^2 - AC = 0$$

En développant, l'équation suivante est obtenue :

$$k_1 x^2 + k_2 y^2 + 2k_3 xy + 2k_4 x + 2k_5 y + k_6 = 0$$

Cette équation est l'équation d'une conique et l'expression de ses coefficients dépend des coefficients de la quadrique :

$$\begin{aligned}
k_1 &= q_7^2 - q_1 q_{10} \\
k_2 &= q_8^2 - q_2 q_{10} \\
k_3 &= q_7 q_8 - q_4 q_{10} \\
k_4 &= q_7 q_9 - q_5 q_{10} \\
k_5 &= q_7 q_7 - q_6 q_{10} \\
k_6 &= q_9^2 - q_3 q_{10}
\end{aligned}$$

Si la matrice $\mathbf{\Gamma}$ (6.3) est décomposée en :

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{\Gamma}_{\mathbf{33}} & \mathbf{\Gamma}_{13} \\ \mathbf{\Gamma}_{13}^\top & q_{10} \end{pmatrix}$$

Alors la relation entre une quadrique et la conique projetée peut s'écrire sous la forme matricielle et non linéaire suivante :

$$\mathbf{C} = (\mathbf{\Gamma}_{13} \mathbf{\Gamma}_{13}^\top - q_{10} \mathbf{\Gamma}_{\mathbf{33}}) \quad (6.10)$$

Les termes des deux côtés de cette égalité sont des matrices symétriques 3×3 , cette équation équivaut donc au plus à 6 égalités indépendantes. Ainsi, il n'est pas possible de retrouver les 10 paramètres d'une quadrique en connaissant une seule projection. D'autre part, cette relation est non-linéaire. Pour estimer les coefficients d'une quadrique à partir de l'estimation de ses coniques projetées, il faut donc recourir à des méthodes d'optimisation numérique non linéaires qui seront exposées dans le paragraphe suivant.

Remarque : cette équation de projection n'est évidemment valide que si les coefficients de la quadrique sont exprimés dans le repère de la caméra courante (voir équation (6.6)).

6.3.2 Dans l'espace dual.

Pour calculer la projection perspective d'une fonction quadratique, une manière élégante de procéder est d'utiliser les propriétés de l'espace dual à la quadrique qui est composé de tous les plans tangents à la quadrique [Ma 96, Cross 98].

Par définition, la quadrique duale \mathcal{Q}^* est telle que $\mathbf{\Gamma}^* = \text{adj}(\mathbf{\Gamma})$, où $\text{adj}(\mathbf{A})$ est la matrice adjointe à \mathbf{A}^1 .

\mathcal{Q}^* est le lieu géométrique de tous les plans $\mathbf{\Pi}$ qui satisfont l'équation quadratique homogène suivante :

$$\mathbf{\Pi}^\top \mathbf{\Gamma}^* \mathbf{\Pi} = 0 \quad (6.11)$$

De la même manière, la conique duale \mathcal{C}^* est telle que $\mathbf{C}^* = \text{adj}(\mathbf{C})$ et \mathcal{C}^* est le lieu géométrique de toutes les lignes \mathbf{l} tangentes à la quadrique telles que :

$$\mathbf{l}^\top \mathbf{C}^* \mathbf{l} = 0 \quad (6.12)$$

¹Soit \mathbf{A} , une matrice carrée d'ordre n , la matrice des cofacteurs X_{ij} des éléments a_{ij} est la matrice adjointe de \mathbf{A} ou co-matrice de \mathbf{A} : $\text{adj}(\mathbf{A}) = \text{com}\mathbf{A} = [X_{ij}] = [(-1)^{i+j} \Delta_{ij}]$, où Δ_{ij} est le déterminant de la matrice \mathbf{A} dont on a supprimé la ligne i et la colonne j . Si \mathbf{A} est inversible alors, il existe une relation directe entre l'inverse de \mathbf{A} et sa co-matrice : $\mathbf{A}^{-1} = \text{adj}(\mathbf{A})/\det(\mathbf{A})$.

Rappelons que la projection \mathbf{x} d'un point \mathbf{X} sur le plan image d'une caméra est donnée par (2.16) : $\mathbf{x} = \mathbf{P}\mathbf{X}$, où \mathbf{P} est la matrice de projection. Ainsi, $\mathbf{P} = \mathbf{A}$ si la quadrique est déjà exprimée dans le repère de la caméra et $\mathbf{P} = \mathbf{A} {}^c\mathbf{M}_b$ si la quadrique est exprimée dans le repère de base.

D'après le paragraphe 2.1.2.3, les coordonnées de la ligne \mathbf{l} passant par le centre optique de la caméra et \mathbf{x} peuvent être défini par l'équation (2.17). En remarquant que :

$$\mathbf{l}^\top \mathbf{x} = \mathbf{l}^\top \mathbf{P}\mathbf{X} = \mathbf{\Pi}^\top \mathbf{X}$$

on obtient finalement la caractérisation de $\mathbf{\Pi}$,

$$\mathbf{\Pi} = \mathbf{P}^\top \mathbf{l}$$

En substituant $\mathbf{\Pi}$ ainsi défini dans l'équation (6.11), on obtient :

$$\mathbf{l}^\top \mathbf{P} \mathbf{\Gamma}^* \mathbf{P}^\top \mathbf{l} = 0$$

Ceci garantit que \mathbf{l} est tangente au contour projeté. En revenant à la définition d'une conique duale (équation (6.12)), on obtient l'égalité suivante :

$$\mathbf{C}^* = \mathbf{P} \mathbf{\Gamma}^* \mathbf{P}^\top \quad (6.13)$$

Ainsi, bien que la relation entre \mathbf{C} et $\mathbf{\Gamma}$ soit non linéaire (6.10), l'équation entre leurs matrices duales \mathbf{C}^* et $\mathbf{\Gamma}^*$ est linéaire (6.13).

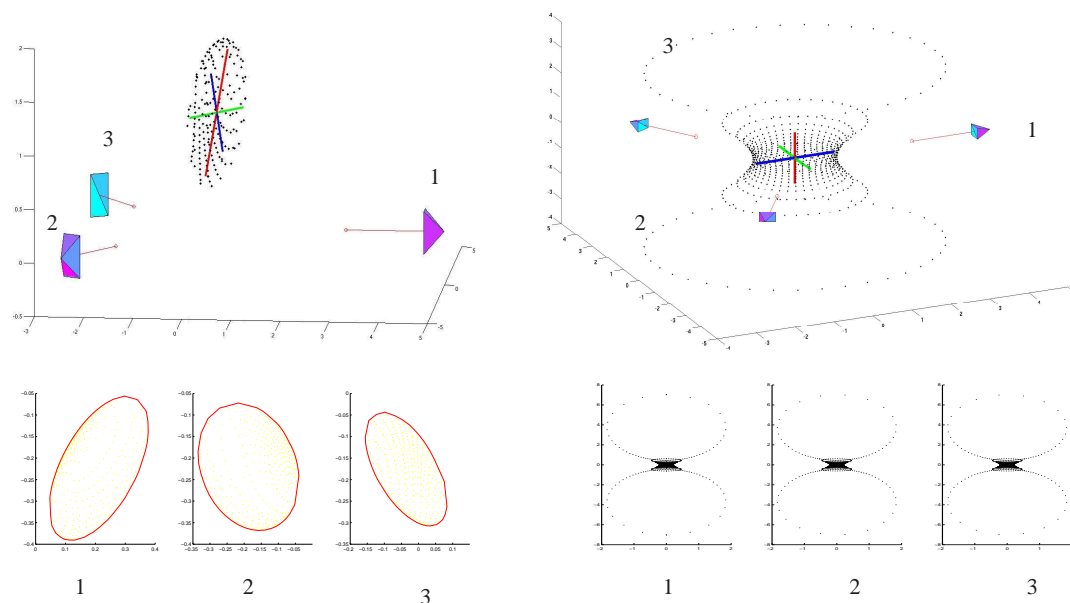


FIG. 6.4 – Exemple de projection d'une quadrique sur trois vues. À gauche, conique projetée à partir d'un ellipsoïde. À droite, coniques projetées à partir d'un hyperboloïde à une nappe.

Les coniques, expressions de la projection d'une quadrique, peuvent être rétro-projetées pour inférer les coefficients de la quadrique.

Estimation des paramètres d'une quadrique

Ce chapitre présente la modélisation proposée pour estimer les paramètres d'une quadrique à partir de la mesure de ses projections dans un ensemble de vues.

Une quadrique en dimension 3 est une fonction implicite à 9 degrés de liberté. Dans le cadre d'une projection perspective non bruitée, il a été montré qu'une quadrique peut être reconstruite par des méthodes déterministes sans ambiguïté à partir de 9 mises en correspondance de points dans deux images de la quadrique [Shashua 97]. Si une contrainte est ajoutée sur l'équation de la quadrique, par exemple la quadrique est dégénérée, alors seules 8 mises en correspondances de points non coplanaires sont requises [Cross 98].

Par ailleurs, la projection d'une quadrique est une conique dont les paramètres imposent 5 contraintes sur les paramètres de la quadrique observée. Ainsi, si une quadrique est observée depuis plusieurs vues, alors il est possible d'estimer ses paramètres à partir de la mesure des coniques projetées dans les images. En particulier, il a été démontré qu'un ellipsoïde peut être reconstruit depuis la mesure de sa projection orthogonale dans trois vues [Ma 94, Karl 94] et qu'une quadrique généralisée peut également être reconstruite à partir de ses projections perspectives dans trois vues [Ma 96].

Une quadrique peut également être reconstruite à partir de la mise en correspondance de points et d'estimations de leurs coniques projetées. Par exemple, les coefficients d'une quadrique peuvent être obtenus à partir de la mesure d'une conique et de 4 mises en correspondance [Shashua 97] ou encore à partir de deux coniques, mesurées dans deux vues, et une mise en correspondance d'un point de surface [Cross 98]. Ces travaux se sont principalement intéressés aux aspects géométriques [Karl 94, Ma 94, Kayikcioglu 00] et peu d'expérimentations ont été faites sur des objets réels [Ma 96, Cross 98]. Dans ce cas, les objets traités sont de forme quadratique parfaite : il s'agit de balles [Ma 96], de bouteilles, de cheminées de refroidissement de centrale nucléaire [Cross 98], respectivement sphères, cylindres et hyperboloïdes.

Les paramètres d'une fonction implicite peuvent être estimés en exprimant la relation entre le mouvement de la caméra et l'évolution des contours occultants dans les images. Ce lien peut être traduit par la matrice d'interaction associée à la primitive considérée [Chaumette 90a, Boukir 93]. La reconstruction est alors le pendant de l'asservissement visuel. Sans stratégie de déplacement de la caméra, ces méthodes sont relativement inefficaces à cause des erreurs sur l'estimation de la vitesse

des primitives dues à l'acquisition des images à des instants discrets. Ces algorithmes donnent de meilleurs résultats si elles sont couplées à une méthode de vision active. Typiquement, la caméra peut être déplacée de façon à garantir que l'estimation de la vitesse des primitives est juste, par exemple, en assurant que cette vitesse est nulle [Boukir 93, Bouteloup 94]. En d'autres termes, en garantissant que le contour occultant de l'objet est constant au cours du déplacement de la caméra. Les erreurs de discrétisation sont ainsi supprimées. [Coudert 97] a prolongé les travaux de [Bouteloup 94] en y ajoutant un terme d'accélération.

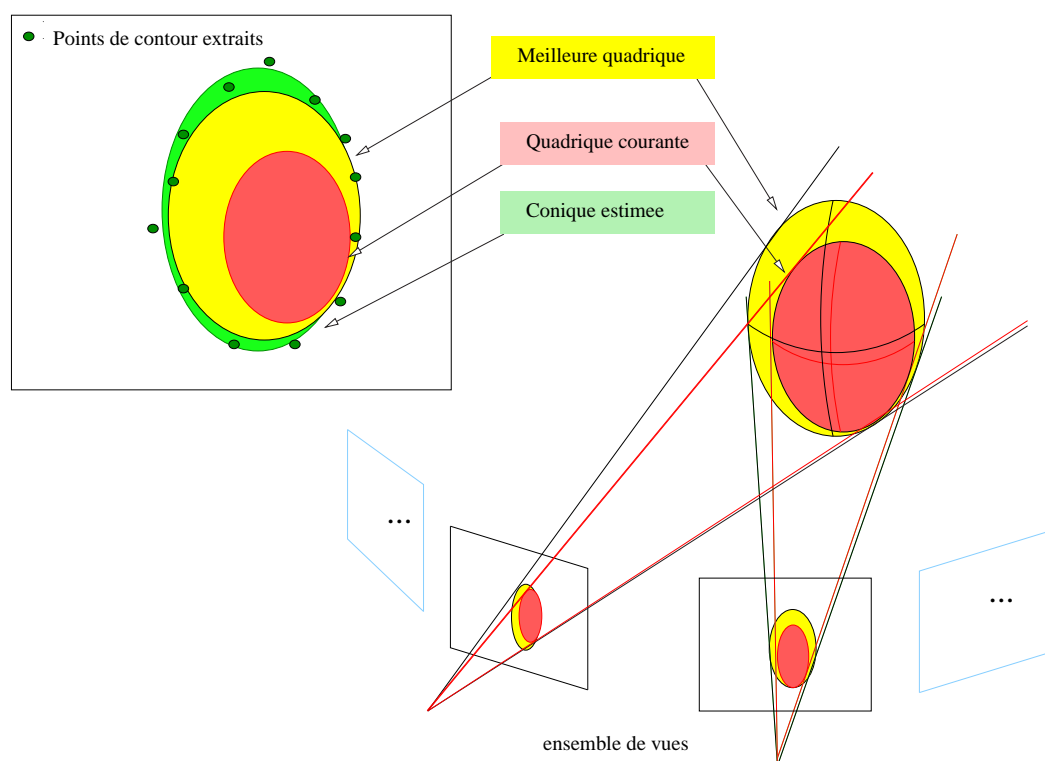


FIG. 7.1 – Estimation des paramètres de la quadrique

Dans cette section, trois méthodes de reconstruction seront présentées (voir figure 7.1). Ces méthodes nécessitent l'estimation des coniques qui correspondent aux contours de l'objet dans les différentes vues. Aussi, le premier paragraphe présente une méthode d'estimation des paramètres d'une conique à partir d'un ensemble de points de contour. Les paragraphes suivants présenteront tour à tour, une méthode linéaire d'estimation des paramètres de la quadrique reposant sur les propriétés de l'espace dual, puis deux méthodes non linéaires d'identification des paramètres reposant sur le lien direct entre les coefficients d'une quadrique et de ses coniques projetées. La troisième est une amélioration de la seconde en utilisant des estimateurs statistiques pour pondérer la contribution de chacune des vues. Dans la suite, la méthode duale servira d'initialisation aux méthodes d'estimation non linéaires.

7.1 Estimation des paramètres d'une conique.

Dans une vue, l'équation de la conique correspondant au mieux à la projection de l'objet est estimée à partir de la détection de ses contours. La segmentation d'un objet sur un arrière plan quelconque est un problème de recherche ouvert. La méthode de traitement d'image que nous utilisons pour détecter les contours de l'objet est détaillée dans le chapitre 8.

Soient $\{\mathbf{x}_i(x_i, y_i, 1)\}$ avec $i \in [1, n]$ un ensemble de n points détectés sur le contour de l'objet. Le problème est de faire correspondre une conique à cet ensemble de n points¹. Rappelons l'équation d'une conique (6.7) :

$$\mathcal{C}(\mathbf{x}) = k_1x^2 + k_2y^2 + 2k_3xy + 2k_4x + 2k_5y + k_6$$

où k_1 , k_2 et k_3 ne peuvent pas être simultanément nuls. Comme les points mesurés proviennent d'un traitement d'image, ils sont nécessairement bruités. Ainsi, il est peu probable de trouver un ensemble de paramètres pour la conique \mathcal{C} tel que $\mathcal{C}(\mathbf{x}) = 0$. Les paramètres seront donc estimés en minimisant une fonction de coût. Pour des raisons d'implémentation, cette fonction est souvent choisie comme la distance algébrique des points à la courbe (d'autres méthodes sont présentées dans [Zhang 97]).

$$f = \sum_{i=1}^n \mathcal{C}^2(\mathbf{x}_i) \quad (7.1)$$

Un minimum immédiat peut être trouvé pour cette équation pour $k_1 = k_2 = k_3 = k_4 = k_5 = k_6 = 0$. Pour éviter de tomber dans ce minimum local, l'équation de la conique doit être normalisée.

Normalisation des coefficients de la conique. Les paramètres d'une conique sont définis à un facteur d'échelle près. Il est nécessaire de rendre la représentation unique pour pouvoir identifier ses paramètres sans tomber dans le minimum qui annule tous ses coefficients.

Plusieurs techniques de normalisation sont détaillées dans [Zhang 97]. Dans le cadre de cette étude, la méthode choisie est une normalisation telle que $\sum_i k_i^2 = 1$. Cette normalisation est générale et n'impose pas de contrainte particulière sur la conique puisque la norme du vecteur des coefficients d'une conique quelconque ne s'annule que lorsque tous les coefficients sont nuls.

Fonction de coût. Soit $\mathbf{k} = (k_1, k_2, k_3, k_4, k_5, k_6)^\top$ le vecteur des coefficients de la conique et $\mathbf{a}_i = (x_i^2, y_i^2, 2x_i, 2y_i, 1)^\top$, le vecteur associé à un point \mathbf{x}_i , l'équation (6.7) au point \mathbf{x}_i peut se réécrire :

$$\mathbf{a}_i^\top \mathbf{k} = 0 \quad \text{avec } \|\mathbf{k}\| = 1$$

et pour n points l'équation s'écrit :

$$\mathbf{A}\mathbf{k} = 0 \quad \text{avec } \mathbf{A} = (\mathbf{a}_1 \dots \mathbf{a}_n)^\top$$

et la fonction à minimiser s'écrit alors :

$$f(\mathcal{C}) = (\mathbf{A}\mathbf{k})^\top (\mathbf{A}\mathbf{k}) = \mathbf{k}^\top \mathbf{B}\mathbf{k}$$

La solution de cette équation est le vecteur propre de \mathbf{B} associé à la plus petite valeur propre.

Critique de la méthode. L'avantage d'utiliser la distance algébrique est sans conteste le gain en efficacité de calcul. Cependant, il arrive parfois que les résultats obtenus soient incorrects [Zhang 97]. Selon la normalisation choisie, la fonction à minimiser n'est généralement pas invariante aux transformations euclidiennes. Typiquement, une normalisation par $k_{10} = 1$ n'est pas invariante aux translations.

La seconde raison est qu'en utilisant cette distance, un point n'a pas la même influence sur l'estimation des paramètres de la quadrique selon sa position dans le plan. En considérant l'équation normalisée d'une conique et en utilisant les mêmes notations que celles de la figure 7.2, la distance algébrique d'un point \mathbf{x}_i à la conique \mathcal{C} est :

$$\mathcal{C}(\mathbf{x}_i) = k_1x_i^2 + k_2y_i^2 + k_6 = -k_6(b^2/c^2 - 1)$$

¹Le lecteur intéressé pourra se référer au tutoriel sur les techniques d'estimation de paramètres [Zhang 97]

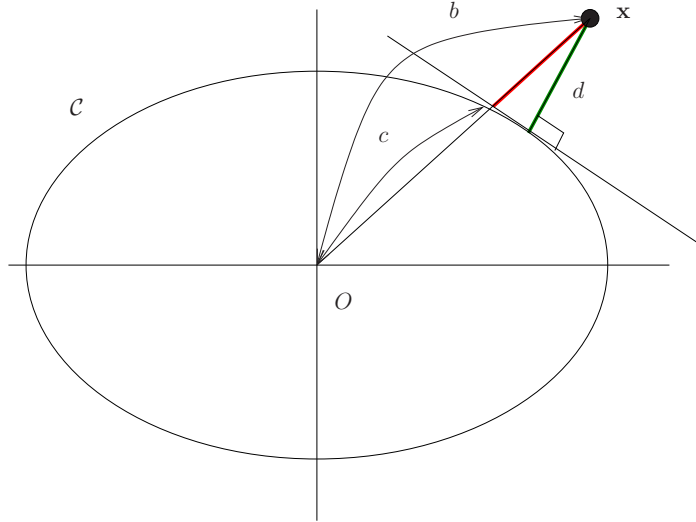


FIG. 7.2 – Distance algébrique ($\propto (b^2/c^2 - 1)$) et euclidienne (d) d'un point \mathbf{x} à une conique C .

où b est la distance séparant le point \mathbf{x} considéré du centre de la conique O . En notant \mathbf{x}_{cross} le point d'intersection entre la conique et le rayon (O, \mathbf{x}_i) , la distance c est la distance séparant O et \mathbf{x}_{cross} . Un point situé sur une zone de forte courbure aura donc moins d'influence qu'un point sur une courbure faible puisque c est plus important dans les zones de forte courbure. Ce problème est connu sous le terme de *biais des fortes courbures* (en anglais, *high curvature bias*).

Pour résoudre ce problème, une idée naturelle est d'utiliser la distance euclidienne séparant le point et la courbe (d sur la figure 7.2). Une méthode permettant de retrouver les coefficients en utilisant la distance euclidienne est présentée dans [Zhang 97]. Le calcul de la distance euclidienne d'un point à une conique est assez complexe et non linéaire. Ainsi estimer les coefficients de la conique en utilisant la distance euclidienne nécessite de recourir à des méthodes d'optimisation non linéaires qui demandent un temps de calcul conséquent en comparaison d'une méthode utilisant la distance algébrique, c'est pourquoi nous utiliserons une méthode algébrique tout en étant conscient du biais engendré par cette méthode. Un point situé dans une région de forte courbure contribuera moins à l'estimation qu'un point situé dans une région de faible courbure.

7.2 Une reconstruction qui exploite les propriétés de l'espace dual

En utilisant l'espace dual, la relation entre la quadrique et ses projections dans les vues depuis lesquelles elle est observée est linéaire (voir le paragraphe 6.3.2).

Système d'équations linéaires Soit p_{ij} les coefficients de la matrice de projection \mathbf{P} de dimension $\dim(P) = 3 \times 4$. Si, pour chaque vue $j \in [1, N]$, on note k_{ji} les paramètres de la conique estimée, alors l'équation (6.13) peut s'écrire sous la forme

$$\mathbf{c}^*_j = \mathbf{A}_j \mathbf{q}^* \quad (7.2)$$

où $\mathbf{c}^*_j = (k^*_{j1} \cdots k^*_{j6})^\top$, $\mathbf{q}^* = (q^*_1 \cdots q^*_{10})^\top$ et

$$\mathbf{A}_j^\top = \begin{pmatrix} p_{11}^2 & p_{21}^2 & p_{11}p_{21} & p_{11}p_{31} & p_{21}p_{31} & p_{31}^2 \\ p_{12}^2 & p_{22}^2 & p_{12}p_{22} & p_{12}p_{32} & p_{22}p_{32} & p_{32}^2 \\ p_{13}^2 & p_{23}^2 & p_{13}p_{23} & p_{13}p_{33} & p_{23}p_{33} & p_{33}^2 \\ 2p_{11}p_{12} & 2p_{21}p_{22} & p_{12}p_{21}+p_{11}p_{22} & p_{12}p_{31}+p_{11}p_{32} & p_{22}p_{31}+p_{21}p_{32} & 2p_{31}p_{32} \\ 2p_{11}p_{13} & 2p_{21}p_{23} & p_{11}p_{23}+p_{13}p_{21} & p_{11}p_{33}+p_{13}p_{31} & p_{21}p_{33}+p_{23}p_{31} & 2p_{31}p_{33} \\ 2p_{12}p_{13} & 2p_{22}p_{23} & p_{12}p_{23}+p_{13}p_{22} & p_{12}p_{33}+p_{13}p_{32} & p_{22}p_{33}+p_{23}p_{32} & 2p_{32}p_{33} \\ 2p_{11}p_{14} & 2p_{21}p_{24} & p_{14}p_{21}+p_{11}p_{24} & p_{14}p_{31}+p_{11}p_{34} & p_{24}p_{31}+p_{21}p_{34} & 2p_{31}p_{34} \\ 2p_{14}p_{12} & 2p_{24}p_{22} & p_{14}p_{22}+p_{12}p_{24} & p_{14}p_{32}+p_{12}p_{34} & p_{24}p_{32}+p_{22}p_{34} & 2p_{34}p_{32} \\ 2p_{13}p_{14} & 2p_{23}p_{24} & p_{13}p_{24}+p_{14}p_{23} & p_{13}p_{34}+p_{14}p_{33} & p_{23}p_{34}+p_{24}p_{33} & 2p_{33}p_{34} \\ p_{14}^2 & p_{24}^2 & p_{14}p_{24} & p_{14}p_{34} & p_{24}p_{34} & p_{34}^2 \end{pmatrix}$$

Cette équation produit six relations linéaires entre les paramètres de la conique duale et ceux de la quadrique duale. Les lignes de \mathbf{A} sont linéairement indépendantes, $\text{rang}(\mathbf{A}) = 6$, cependant, les coefficients des coniques duales sont obtenus à un facteur près. Ainsi, étant données une vue et la conique duale estimée pour cette vue, l'équation (7.2) impose seulement cinq contraintes indépendantes et linéaires sur la quadrique duale. La solution de l'équation (7.2) est une famille de quadriques duales.

Facteur d'échelle. Pour combiner les équations de plusieurs vues, il faut introduire un facteur d'échelle α_i associé à chaque vecteur des coefficients de la conique duale. Pour N vues, l'équation (7.2) peut donc être réécrite de la manière suivante :

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{c}^*_1 & 0 & \cdots & 0 \\ \mathbf{A}_2 & 0 & \mathbf{c}^*_2 & & 0 \\ \vdots & 0 & 0 & \ddots & \vdots \\ \mathbf{A}_N & 0 & 0 & \cdots & \mathbf{c}^*_N \end{pmatrix} \begin{pmatrix} \mathbf{q}^* \\ \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} = 0 \quad (7.3)$$

$$\mathbf{A}\mathbf{b} = 0 \quad (7.4)$$

Résolution du système d'équations. Pour $N > 3$ vues, \mathbf{q}^* est obtenu par une méthode des moindres carrés. La solution \mathbf{b} appartient au noyau de \mathbf{A} . Cette solution peut être trouvée à partir d'une décomposition en valeurs singulières de \mathbf{A} . L'algorithme est décrit plus en avant dans l'encadré 7.3.

7.3 Une reconstruction basée sur la relation directe entre une quadrique et ses projections

Ce paragraphe présente une méthode d'estimation non linéaire des paramètres de la quadrique basées sur l'équation directe (6.10).

Les paramètres d'une conique résultant de la projection d'une quadrique sur le plan image d'une caméra dépendent des paramètres de la quadrique exprimés dans le repère de la caméra (voir équation (6.10)), ou encore de la position de la caméra et des paramètres de la quadrique exprimés dans un référentiel fixe, par exemple \mathcal{R}_b . Ce sont ces paramètres qui sont à estimer.

Fonction de coût. L'erreur à minimiser peut être considérée comme une mesure de la distance entre les données expérimentales et le modèle théorique qui prédit ces données, que sont respectivement les paramètres des coniques estimées dans les plans images aux différentes positions de la caméra embarquée et les paramètres des coniques calculées par projection de la quadrique estimée.

La fonction de coût $f(\mathcal{Q})$ correspond donc à la différence entre les coefficients des coniques estimées dans les vues et ceux des coniques projetées à partir de l'équation courante de la quadrique. Cela revient

Reconstruction Duale

Une quadrique peut être reconstruite par une méthode des moindres carrés linéaires en passant par son espace dual. Trois vues ou plus sont alors nécessaires pour estimer les 9 paramètres indépendants de la quadrique \mathcal{Q} .

L'algorithme de calcul linéaire est le suivant :

1. **Acquisition :**
 $N \geq 3$ vues sont acquises, les coefficients des \mathbf{c}_j sont estimés et permettent de calculer les coefficients de la conique duale correspondants \mathbf{c}_j^* .
2. **Calcul de \mathbf{A} :**
 Pour chaque vue la matrice \mathbf{A}_j est calculée en fonction de la position occupée par la caméra. La matrice \mathbf{A} est ensuite construite en utilisant les \mathbf{A}_j et les \mathbf{c}_j^* .
3. **Calcul de la quadrique duale :**
 Les paramètres de la quadrique qui correspond le mieux aux projections observées sont calculés à partir d'une décomposition en valeur singulière de $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}$. \mathbf{U} et \mathbf{V} sont deux matrices unitaires et \mathbf{S} est une matrice rectangulaire, de mêmes dimensions que \mathbf{A} , et qui présentent des termes positifs sur sa diagonale. En notant \mathbf{v}_i , les vecteurs colonnes de \mathbf{V} et s_i les valeurs de la diagonale de \mathbf{S} , c'est-à-dire ses valeurs singulières. Les colonnes \mathbf{v}_i de \mathbf{V} qui correspondent à des valeurs singulières s_i nulles forment une base orthonormée du noyau de \mathbf{A} . La solution \mathbf{b} est une combinaison linéaire de ces vecteurs.
4. **Calcul de la quadrique :**
 Les coefficients de la quadrique \mathbf{q} sont estimés à partir des coefficients de la quadrique duale \mathbf{q}^* obtenus à l'étape précédente.

FIG. 7.3 – *Reconstruction duale.*

en fait à minimiser les surfaces jaunes sur la figure 7.1 de sorte à ce que les projections de la quadrique réelle et celles de la quadrique estimée se superposent.

$$f(\mathcal{Q}) = \sum_j^N \sum_i^6 \left(\widehat{k}_{ij} - k_{ij}(\mathcal{Q}, {}^b\mathbf{M}_{c_j}) \right)^2 = \sum_j^N \sum_i^6 r_{ij}^2(\mathcal{Q}) \quad (7.5)$$

où $r_{ij}^2(\mathcal{Q})$ sont les résidus au modèle, c'est-à-dire les écart entre les coniques estimées \widehat{k}_{ij} et le modèle $k_{ij}(\mathcal{Q}, {}^b\mathbf{M}_{c_j})$.

Si une estimation de l'écart type σ_{ij} du bruit qui affecte chaque mesure est disponible alors il est utilisé pour pondérer la contribution de la mesure au chi-carré. Une mesure aura d'autant plus de poids que son incertitude sera faible.

$$\chi^2(\mathcal{Q}) = \sum_j^N \sum_i^6 \frac{\left(\widehat{k}_{ij} - k_{ij}(\mathcal{Q}, {}^b\mathbf{M}_{c_j}) \right)^2}{\sigma_{ij}} = \sum_j^N \sum_i^6 w_{ij} \left(\widehat{k}_{ij} - k_{ij}(\mathcal{Q}, {}^b\mathbf{M}_{c_j}) \right)^2 \quad (7.6)$$

où \widehat{k}_{ij} est le i ème paramètre, $i \in [1, 6]$, de la conique estimée à partir des contours extraits dans la j ème vue, $j \in [1, N]$, et $k_{ij}(\mathcal{Q}, {}^b\mathbf{M}_{c_i})$ est le i ème paramètre de la conique calculée à partir de la projection de la quadrique \mathcal{Q} dans la j ème vue. σ_{ij} est un terme de normalisation qui permet d'obtenir une loi chi-carré. À l'instant initial $\sigma_{ij} = 1$ et à la fin du processus d'optimisation $\sigma_{ij} = \sum_j^N \sum_i^6 \left(\widehat{k}_{ij} - k_{ij}(\mathcal{Q}_{min}, {}^b\mathbf{M}_{c_j}) \right)^2$. Cette fonction de coût peut être réécrite comme la norme d'un vecteur \mathbf{F} de dimension $\dim(\mathbf{F}) = 6 \times N$ et dont les coefficients sont les $f_{ij} = \widehat{k}_{ij} - k_{ij}(\mathcal{Q}, {}^b\mathbf{M}_{c_i})$.

Normalisation des paramètres de la conique \mathcal{C} . Comme le montre l'équation (6.7), une conique est définie à un facteur multiplicatif près. Pour pouvoir comparer les paramètres estimés \widehat{k} et les paramètres projetés k il faut lever cette ambiguïté. Pour ne pas perdre en généralité et pouvoir considérer tous les types de coniques, la conique sera rendue unique par une normalisation identique à celle proposée dans le paragraphe 7.1. Les coefficients k_{ij} de la conique estimée dans la vue j deviennent alors $\frac{k_{ij}}{\sqrt{\sum_i k_{ij}^2}}$.

Estimation des paramètres de la quadrique \mathcal{Q} . Le problème de l'identification des paramètres \mathcal{Q} est un problème de minimisation non linéaire d'une fonction de coût f définie par l'équation (7.6). Ce type de problème peut être résolu par une méthode des moindres carrés non linéaire [Bonnas 97].

Connaissant les valeurs courantes des paramètres de la quadrique, la méthode de minimisation permet d'obtenir un nouveau jeu de paramètres qui améliore l'estimation. Ainsi, le processus d'optimisation est itératif et sera répété jusqu'à ce que la valeur de la fonction de coût n'évolue plus ou bien trop peu au regard d'un seuil fixé arbitrairement.

L'utilisation d'une méthode des moindres carrés est particulièrement intéressante puisque la fonction de coût f peut alors être approximée par une forme quadratique au voisinage de l'optimum :

$$\chi^2 \sim \gamma - \mathbf{d}\mathbf{q} + \frac{1}{2}\mathbf{q}^\top \mathbf{H}\mathbf{q}$$

où \mathbf{H} est le Hessien de la fonction de coût f , \mathbf{q} est le vecteur des paramètres de la quadrique.

1. Si l'approximation quadratique est vraie, alors il est possible de passer des paramètres courants aux paramètres optimaux en un seul pas :

$$\mathbf{q}_{min} = \mathbf{q} - \mathbf{H}^{-1}\nabla\mathbf{F}(\mathbf{q})$$

2. Sinon, un pas, dont la valeur est fixée arbitrairement, est effectué dans la direction du gradient de f (méthode des plus fortes pentes ou *steepest descent method*), autrement dit :

$$\mathbf{q}_{t+1} = \mathbf{q}_t - \alpha \nabla \mathbf{F}(\mathbf{q})$$

Pour utiliser ces équations, il faut calculer le Gradient de f et son Hessian qui peuvent tous les deux être approximés à partir du Jacobien de f . Le Jacobien de la fonction de coût f peut être représenté sous la forme d'une matrice \mathbf{J}_C de dimension $\dim(\mathbf{J}_C) = (6 \times N, 10)$, avec N le nombre de vues disponibles.

$$\mathbf{J}_C = \begin{pmatrix} \frac{\partial k_{11}}{q_1} & \frac{\partial k_{11}}{q_2} & \dots & \frac{\partial k_{11}}{q_{10}} \\ \frac{\partial k_{21}}{q_1} & \frac{\partial k_{21}}{q_2} & \dots & \frac{\partial k_{21}}{q_{10}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial k_{61}}{q_1} & \frac{\partial k_{61}}{q_2} & \dots & \frac{\partial k_{61}}{q_{10}} \\ \frac{\partial k_{12}}{q_1} & \frac{\partial k_{12}}{q_2} & \dots & \frac{\partial k_{12}}{q_{10}} \\ \frac{\partial k_{22}}{q_1} & \frac{\partial k_{22}}{q_2} & \dots & \frac{\partial k_{22}}{q_{10}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial k_{6N}}{q_1} & \frac{\partial k_{6N}}{q_2} & \dots & \frac{\partial k_{6N}}{q_{10}} \end{pmatrix} \quad (7.7)$$

Le gradient ∇f et le hessian $\nabla^2 f$ de la fonction de coût sont alors définis par :

$$\begin{aligned} \nabla f &\sim \mathbf{J}_C^\top \mathbf{F} \\ \nabla^2 f &\sim \mathbf{J}_C^\top \mathbf{J}_C + \sum_{i=1,6N} \mathbf{F}_i \mathbf{H}_i \end{aligned}$$

Au voisinage de l'optimum, $\|\mathbf{F}\|$ tend vers zéro, le terme $\sum_{i=1}^{6N} \mathbf{F}_i \mathbf{H}_i$ peut donc être négligé devant $\mathbf{J}_C^\top \mathbf{J}_C$. En pratique, ce terme peut être négligé pour d'autres raisons : si l'erreur de l'approximation quadratique est faible, alors ce terme représente l'erreur de mesure. Cette erreur est décorrélée du modèle. Ainsi, le second terme tend à s'annuler lorsque la somme sur tous les coefficients i est effectuée. L'approximation du Hessian $\mathbf{G} = \mathbf{J}_C^\top \mathbf{J}_C$ obtenue est appelée la *matrice de Gauss-Newton*. Elle est valide au voisinage de l'optimum.

La méthode de Levenberg Marquardt permet de passer continûment du schéma d'inversion du hessian à celui des plus fortes pentes en faisant varier un paramètre λ . Lorsque l'estimation courante des paramètres est loin de l'optimum, un pas est effectué dans la direction du gradient. Au fur et à mesure de l'approche de l'optimum, la méthode passe continuellement vers un Gauss-Newton.

$$\mathbf{q}_{t+1} = \mathbf{q}_t - (\mathbf{G}(\mathbf{q}_t) + \lambda \mathbf{D}(\mathbf{q}_t))^{-1} \mathbf{J}_C(\mathbf{q}_t)^\top \mathbf{F}(\mathbf{q}_t)$$

où \mathbf{D} est une matrice diagonale construite telle que $\mathbf{D}_{ii} = \mathbf{G}_{ii}$. Lorsque λ est grand, $(\mathbf{G}(\mathbf{q}_t) + \lambda \mathbf{D}(\mathbf{q}_t))$ est une matrice dont les valeurs de la diagonale sont dominantes. Ainsi, la méthode revient à une descente du gradient. À l'opposé, lorsque λ tend vers zéro alors la méthode est une méthode de Gauss-Newton.

Une fois qu'un minimum acceptable $f(\mathbf{q}_{min})$ a été trouvé, la variation de f autour de ce minimum pour une variation $\delta \mathbf{q}$ des paramètres ajustés, et avec $\nabla \mathbf{F}(\mathbf{q}_{min}) = 0$, est donnée par :

$$f = f(\mathbf{q}_{min}) + \delta \mathbf{q}^\top \mathbf{G} \delta \mathbf{q} \quad (7.8)$$

La matrice $\mathbf{C} = \sigma^2 \mathbf{G}^{-1} = \sigma^2 (\mathbf{J}^\top \mathbf{J})$ peut être évaluée. C'est une estimation de la covariance \mathbf{C} des erreurs associées aux paramètres \mathbf{q} . Elle représente l'incertitude liée au jeu de paramètres estimés. Plus les valeurs propres de \mathbf{C} sont faibles et mieux les paramètres sont estimés (voir chapitre 9). La distribution de probabilité de la solution inverse est une distribution Gaussienne de covariance $\mathcal{N}(\mathbf{q}_{min} - \mathbf{q}, \mathbf{C})$. Nous verrons que cette matrice sera très utile pour quantifier la qualité de l'estimation et l'information apportée par chacune des vues (voir 9.1).

Algorithme de Levenberg Marquardt

L'algorithme d'optimisation non linéaire permettant l'estimation d'un jeu de paramètres se déroule comme suit :

1. **Acquisition :**
 $N > 3$ images de l'objet sont acquises pour N positions données et les paramètres \hat{k}_{ij} sont estimés.
2. **Initialisation :**
 \mathcal{Q}_0 est initialisée par un jeu de paramètres, qui peuvent être choisis arbitrairement (typiquement une sphère centrée sur la position estimée de l'objet) ou bien calculés lors d'une première étape de calcul linéaire, par exemple en passant par la représentation duale.
3. **Boucle d'estimation :**
 Tant que les critères d'arrêt ne sont pas vérifiés
 - (a) Pour la première itération, on choisit un λ assez petit, par exemple $\lambda = 0.001$ et un $v > 1$
 - (b) Calcul du jacobien $\mathbf{J}_C(\mathbf{k}_t)$
 - (c) Calcul de la fonction de coût $\mathbf{F}(\mathbf{k}_t)$
 - (d) Estimation des paramètres $\mathbf{q}_{temp} = \mathbf{q}_t - (\mathbf{G} + \lambda \mathbf{D})^{-1} \mathbf{J}_C^T \mathbf{F}$
 - si $\mathbf{q}_{temp} \geq \mathbf{q}_t$ alors augmenter λ d'un facteur v : $\lambda = v\lambda$ et retourner à (d)
 - si $\mathbf{q}_{temp} < \mathbf{q}_t$ alors mettre à jour les paramètres courants $q_t = q_{temp}$ et diminuer λ d'un facteur v : $\lambda = \lambda/v$ et passer en (e)
 - (e) Calcul des critères d'arrêt en fonction de (\mathbf{k}_t) . Nous utiliserons deux critères d'arrêt :
 - Résolution du système d'équation non linéaire :
 si la norme euclidienne $\|\mathbf{F}\| < S_F$ est en deçà d'un seuil déterminé arbitrairement, arrêter le processus.
 - Condition de stationnarité :
 si la norme euclidienne du gradient de la fonction de coût $\|\mathbf{J}_C^T \mathbf{F}\| < S_\nabla$, arrêter le processus. Elle équivaut à une faible variation des paramètres, typiquement, $S_\nabla = 10^{-3}$.
 - Aucun des critères n'est rempli, retourne en (b)

FIG. 7.4 – Algorithme de Levenberg Marquardt. [Marquardt 63]

7.4 Une reconstruction robuste

L'estimation est dite robuste si elle ne perd pas trop de ses qualités optimales lorsqu'on s'éloigne des hypothèses sous lesquelles le modèle a été conçu. Dans le cas de la reconstruction d'un objet réel, des erreurs apparaissent dans l'estimation des coniques et dans la position des vues. Il peut donc arriver que certaines vues donnent de mauvaises mesures qui perturbent l'estimation au lieu de l'améliorer.

Pour parer à ce problème, si la géométrie épipolaire peut être calculée, elle peut être utilisée pour rectifier les coefficients d'une conique estimée dans une vue à partir des coefficients d'une conique de référence [Cross 98]. Cette méthode demande de désigner une projection comme *conique de référence* à laquelle toutes les autres projections sont ajustées. Par conséquent, si la conique de référence est mal estimée, l'estimation des paramètres de la quadrique est biaisée. Aussi, plutôt que de désigner arbitrairement une conique de référence, nous proposons d'utiliser des estimateurs robustes, les M-Estimateurs [Beaton 74], pour pondérer les contributions des différentes vues et donner moins de poids aux vues qui présentent une erreur de mesure importante.

Les M-Estimateurs. Les M-Estimateurs ont pour objet de réduire l'influence des mesures aberrantes en modifiant la fonction de coût. Le principe de l'algorithme des moindres carrés pondérés itérés, ou IRLS (Iteratively Reweighted Least Squares) est rappelé dans [Malis 05] et l'algorithme est présenté dans l'encadré 7.5. Après chaque itération du processus des moindres carrés, un poids est accordé à chaque résidu en considérant son éloignement de la loi de probabilité déterminé par l'ensemble des résidus. Finalement la fonction de coût devient $\mathbf{F}_r = \mathbf{W}\mathbf{F}$ et le *Jacobien pondéré* $\mathbf{J}_r = \mathbf{W}\mathbf{J}$ avec \mathbf{W} une matrice diagonale pour laquelle chaque élément non nul w_i est calculé de la manière suivante :

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}$$

où $\psi(\delta_i/\sigma) = \frac{\partial}{\partial \mathbf{r}} f(\delta_i/\sigma)$, ψ est la fonction d'influence et δ_i est le résidu normalisé donné par $\delta_i = r_i - \text{Med}(\mathbf{r})$, où $\text{Med}(\mathbf{r})$ correspond à la valeur médiane des résidus. Le paramètre σ représente la valeur de l'écart type du bruit sur les mesures.

Fonction d'influence. Parmi les fonctions d'influence qui existent dans la littérature (Cauchy, Geman, McLure), nous avons choisi la fonction de Tukey [Beaton 74] qui rejette complètement les données aberrantes et leur donnent un poids nul. Par exemple, si dans une vue, l'extraction des contours est fautive à cause d'un défaut d'illumination alors que dans les autres vues, cette extraction de contours est juste, un poids proche de zéro doit être donné à la contribution de cette vue. Dans le cas de l'exemple de simulation avec des données bruitées présenté précédemment, cela devrait typiquement permettre de supprimer la contribution de la première vue.

La fonction d'influence de Tukey est la suivante :

$$\psi(u) = \begin{cases} u(S^2 - u^2)^2, & \text{si } |u| \leq S \\ 0, & \text{sinon} \end{cases}$$

où le facteur de proportionnalité pour la fonction de Tukey est $S = 4.6851$. Ce facteur représente une efficacité de 95% dans le cas du bruit Gaussien.

Calcul de l'écart type. Afin d'obtenir une fonction robuste et d'améliorer la précision de détection des données aberrantes, il est nécessaire de définir un degré de confiance des mesures ou valeurs du bruit. Ceci revient à déterminer l'écart type du bruit sur les bonnes mesures.

Pour estimer cet écart type σ , en présence de données aberrantes, nous utilisons la valeur médiane de la déviation absolue, ou *Median Absolute Derivation*. C'est un estimateur robuste de l'écart type des bonnes valeurs. Il est donné par :

$$\hat{\sigma} = 1.48 \text{Med}_i(|\delta_i - \text{Med}(\delta)|) \quad (7.9)$$

Le facteur 1.48 permet d'avoir un estimateur non biaisé dans le cas de données Gaussiennes.

Algorithme des moindres carrés pondérés itérés IRLS (Iteratively Reweighted Least Squares)

L'algorithme est le suivant :

1. **Acquisition** :
(voir l'encadré 7.4)
2. **Initialisation** :
(voir l'encadré 7.4)
3. **Boucle d'estimation** :
Tant que les critères d'arrêt ne sont pas vérifiés
 - (a) Pour la première itération, on choisit un λ assez petit, par exemple $\lambda = 0.001$
 - (b) Calcul du jacobien $\mathbf{J}_C(\mathbf{k}_t)$
 - (c) Calcul de la fonction de coût $\mathbf{F}(\mathbf{k}_t)$
 - (d) Calcul de la matrice des pondérations \mathbf{W}
 - (e) Estimation des paramètres $\mathbf{q}_{temp} = \mathbf{q}_t - \alpha_t (\mathbf{W}\mathbf{G})^{-1} \mathbf{J}_C^T \mathbf{W}\mathbf{F}$
 α est adapté à chaque pas en utilisant une méthode des régions de confiance
 - (f) Calcul des critères d'arrêt en fonction de (\mathbf{k}_t) . (voir l'encadré 7.4)

FIG. 7.5 – Algorithme des moindres carrés pondérés itérés IRLS (Iteratively Reweighted Least Squares).

Calcul des résidus. Dans le cadre d'une estimation des paramètres d'une quadrique à partir des estimations de ses coniques projetées, un poids est calculé par vue, puisque les variables ne sont pas indépendantes. Deux solutions sont envisageables, la première consiste à calculer les résidus comme les $r_j = \sum_{i=1}^6 \left(\hat{k}_{ij} - k_{ij}(\mathcal{Q}, {}^b \mathbf{M}_{c_j}) \right)^2$ avec $j \in [1, N]$. Les M-Estimeurs sont calculés sur N valeurs de résidu. La seconde solution est de calculer les résidus comme les $r_j = \left(\hat{k}_{ij} - k_{ij}(\mathcal{Q}, {}^b \mathbf{M}_{c_j}) \right)^2$ avec $j \in [1, 6N]$ et d'attribuer ensuite à chaque vue le poids le plus faible parmi les six calculés.

7.5 Résultats de simulation

Pour valider les méthodes proposées et comparer leur efficacité, nous les avons testées sur les données simulées suivantes :

1. La quadrique à reconstruire est un ellipsoïde de dimensions et de pose quelconque. Les vues sont réparties sur une sphère de vision. Les données sont non bruitées.
2. Les conditions sont identiques à 1. mais la projection de la quadrique est bruitée dans une vue.

3. Les conditions sont identiques à 1. mais avec des positions de la caméra proches les unes des autres.
4. L'objet à reconstruire n'est pas une quadrique. Les vues sont bien réparties.

7.5.1 Reconstruction non bruitée d'une quadrique

Dans un premier temps, considérons une quadrique quelconque (non centrée et non alignée avec le repère fixe), par exemple $\mathbf{q} = (1, 2, 3, 0, \dots, 0, -1)^\top$ à laquelle une rotation de $\mathbf{R} = (80, 40, 10)^\top$ et une translation de $\mathbf{T} = (1, 2, 0)^\top$ sont appliquées, notons ses paramètres \mathcal{Q}_r . Cet objet est observé depuis six vues. Cinq d'entre elles sont également réparties sur un cercle inclus dans le plan $Z = 0$ et centré sur $\mathbf{X} = (0, 0, 0)^\top$. La dernière observe la scène depuis le dessus (voir la figure 7.6). L'objet est projeté dans les différentes vues et les coniques correspondant au contour de l'objet sont estimées (voir figure 7.6, c).

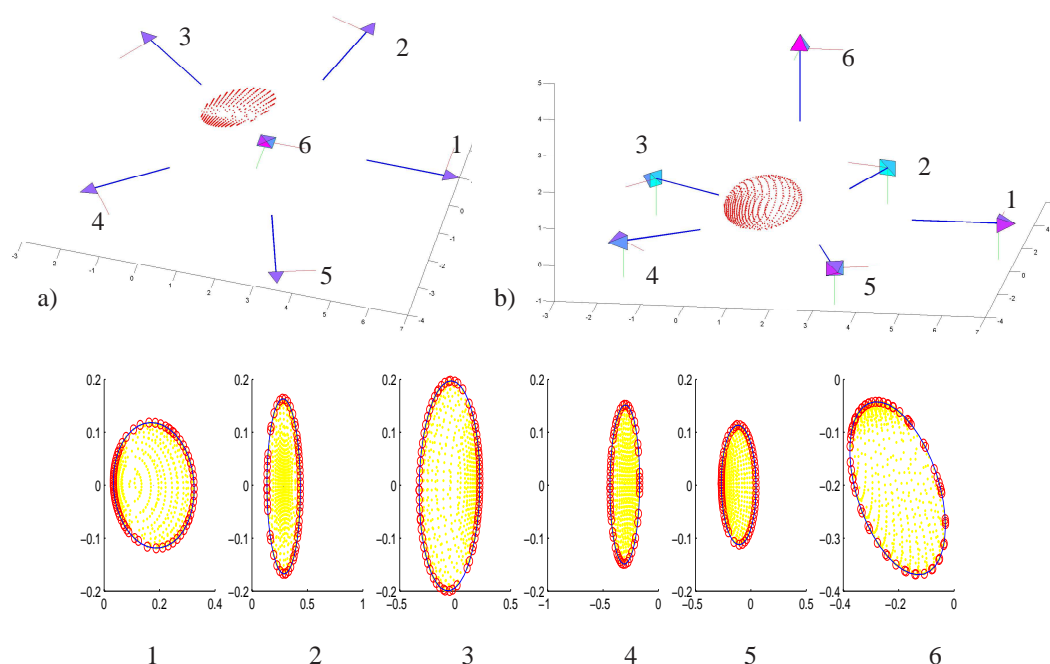


FIG. 7.6 – Simulation non bruitée : une quadrique \mathcal{Q}_r (en rouge) est observée depuis six points de vues (pyramides) répartis sur une sphère de vision. Le système est considéré non bruité. a) Vue du dessus. b) Vue de côté c) Projections dans les plans images. Les points jaunes sont les projections des points de l'objet, les cercles rouges sont les points de contours détectés et les ellipses bleues sont les ellipses estimées.

Les paramètres obtenus par les différents algorithmes seront respectivement notés \mathcal{Q}_d pour la méthode duale, \mathcal{Q}_{nl} pour la méthode non linéaire et \mathcal{Q}_{mest} , pour la méthode non linéaire robuste. Les méthodes non linéaires sont initialisées par un jeu de paramètres arbitraires, par exemple, une sphère de rayon 1 centrée sur l'origine du repère fixe ou sur la position estimée de l'objet. Les critères d'arrêt sont fixés tels que la stationnarité est 10^{-3} et la précision 10^{-3} . Le tableau 7.1 résume les résultats de reconstruction obtenus.

La figure 7.7 présente la reconstruction duale obtenue (en vert) superposée à l'objet initial (en rouge). En l'absence de bruit sur les mesures des points de contours, et dans le cas où l'objet est une

quadrique parfaite, les résultats des méthodes non linéaires sont similaires aux résultats de l'optimisation linéaire duale \mathcal{Q}_d , à cela près que la convergence demande plusieurs itérations.

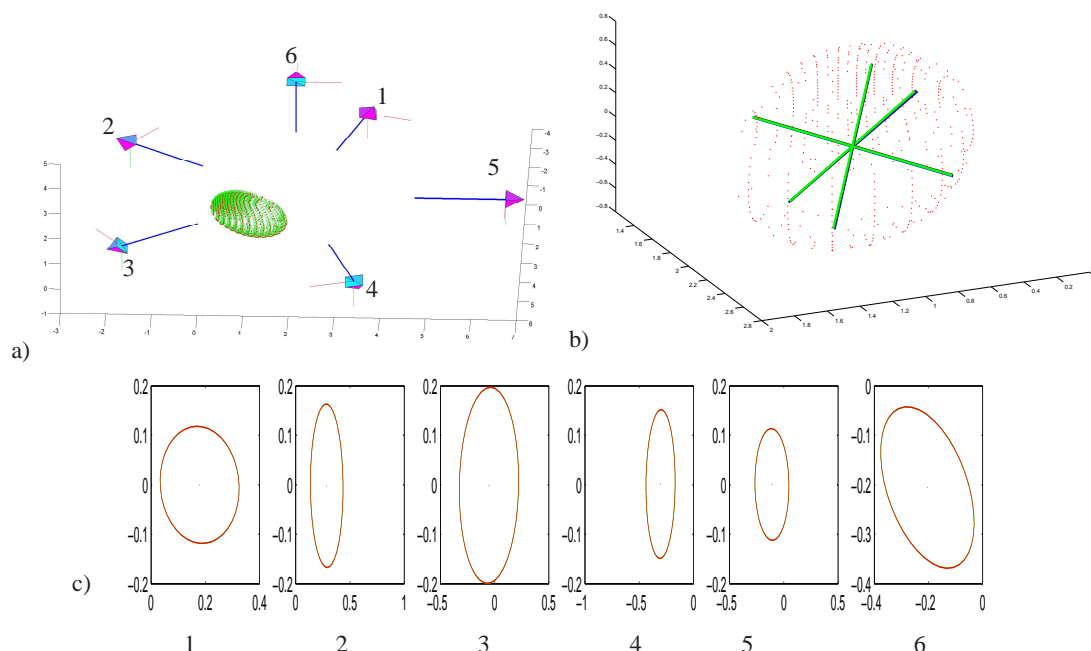


FIG. 7.7 – Simulation non bruitée : résultats de reconstruction d'une quadrique à partir de ses projections en utilisant l'espace dual. a) Vue de dessus, b) Vue de côté, c) Projection dans les vues. La quadrique estimée se projette parfaitement (en vert) sur les coniques estimées (points rouges).

méthode	Dimensions			Centre			Orientation			résidu	erreur sur \mathcal{Q}	nombre d'it.
	d_i	d_j	d_k	X_c	Y_c	Z_c	θ_i	θ_j	θ_k			
\mathcal{Q}_r	1	0,71	0,58	1	2	0	80	40	10			
\mathcal{Q}_d	1,00	0,70	0,58	1,00	2,00	0,00	80,00	40,00	10,00	$2 \cdot 10^{-4}$	$4,2 \cdot 10^{-6}$	1
\mathcal{Q}_{nl}	0,99	0,70	0,57	0,98	1,99	0,00	80,00	40,17	10,03	$3 \cdot 10^{-5}$	$1,0 \cdot 10^{-4}$	12
\mathcal{Q}_{mest}	0,97	0,69	0,56	0,99	1,96	0,00	80,00	40,45	10,26	$2 \cdot 10^{-4}$	$7,7 \cdot 10^{-4}$	18

TAB. 7.1 – Comparaison des méthodes de reconstruction dans un cas non bruité

7.5.2 Reconstruction bruitée d'une quadrique

Considérons l'exemple présenté précédemment, c'est-à-dire une quadrique quelconque observée depuis 6 vues en perturbant faiblement, puis fortement l'estimation de la conique projetée dans une des vues. Dans la première simulation, un bruit gaussien additif de variance relativement faible 0.01 et de moyenne 0.02 perturbe la position des points de contour dans la première vue. Dans la seconde, la variance du bruit est plus importante et vaut 0.1, sa moyenne est de 0.2. La figure 7.8 présente l'estimation des coniques dans les six vues dans le cas non bruité et dans les deux cas bruités.

Bruit faible. Dans le cas d'une perturbation faible, les trois méthodes proposées convergent vers la solution. Les trois reconstructions et les projections résultantes sont présentées dans la figure 7.9. Les résultats sont détaillés dans le tableau 7.2. Les méthodes non linéaires donnent de meilleurs résultats

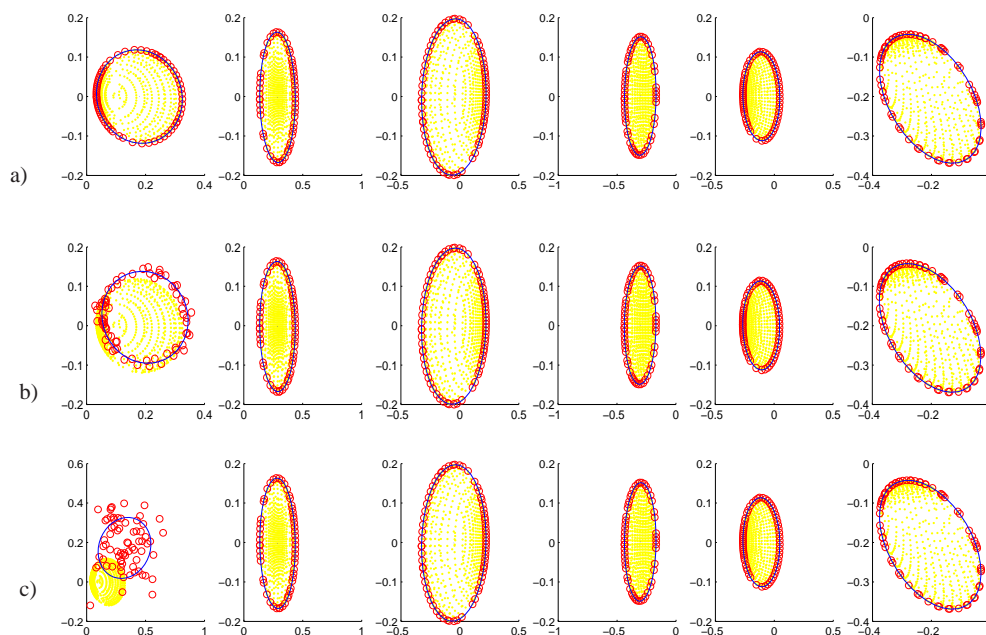


FIG. 7.8 – Projection de la quadriques \mathcal{Q}_r dans 6 vues. a) Aucun bruit n'est ajouté. Dans les deux images suivante, les positions des points de contours de la première vue sont bruitées par un bruit additif gaussien a) de variance 0.01 et de moyenne 0.02 b) de variance 0.1 et de moyenne 0.2. Dans le second cas, l'estimation de la conique dans la première vue apporte une information totalement fausse.

que la méthode duale qui parvient cependant en une seule itération à une bonne estimation de la quadrique englobant l'objet. Après dix itérations, les M-Estimeurs rejettent la première vue (voir 7.10). L'estimation des paramètres de la quadrique s'effectue donc à partir des cinq vues restantes. En calculant, à convergence, la somme des résidus de ces cinq vues pour les deux méthodes non linéaires, on observe de meilleurs résultats pour la méthode robuste.

	Dimensions			Centre			Orientation			résidu	erreur sur \mathcal{Q}	nombre d'it.
	d_i	d_j	d_k	X_c	Y_c	Z_c	θ_i	θ_j	θ_k			
\mathcal{Q}_r	1	0,71	0,58	1	2	0	80	40	10			
\mathcal{Q}_d	1.1	0.70	0.44	1.02	2.05	-0.04	68.84	40.88	5.47	$7,8 \cdot 10^{-2}$	$7,9 \cdot 10^{-2}$	1
\mathcal{Q}_{nl}	0.99	0.71	0.58	0.98	1.99	-0.01	80.42	40.04	9.61	$7,2 \cdot 10^{-4}$ ($4 \cdot 10^{-4}$)	$7,1 \cdot 10^{-5}$	10
\mathcal{Q}_{mest}	0.97	0.68	0.56	0.99	1.98	0.01	80.10	40.17	9.92	$7,2 \cdot 10^{-4}$ ($9 \cdot 10^{-5}$)	$2,1 \cdot 10^{-5}$	20

TAB. 7.2 – Comparaison des méthodes de reconstruction dans un cas bruité : un bruit gaussien de variance 0.01 perturbe les positions des points de contours dans la première vue. Les valeurs entre parenthèses correspondent aux résidus calculés sans la première vue.

Bruit important. Dans ce second exemple, un bruit important perturbe l'estimation d'une conique dans une des vues. Les reconstructions obtenues par les méthodes duale, non linéaire et robuste sont illustrées sur la figure 7.11. Le tableau 7.3 résume les résultats.

La méthode duale donne un résultat aberrant : un hyperboloïde tandis que les méthodes non linéaires convergent vers une solution acceptable (une ellipsoïde). La méthode robuste est particulièrement bien adaptée à ce type de situation : après 4 itérations, le poids associé à la première vue chute à zéro et cette vue n'est jamais réhabilitée par la suite (voir la figure 7.12,c)). L'estimation des

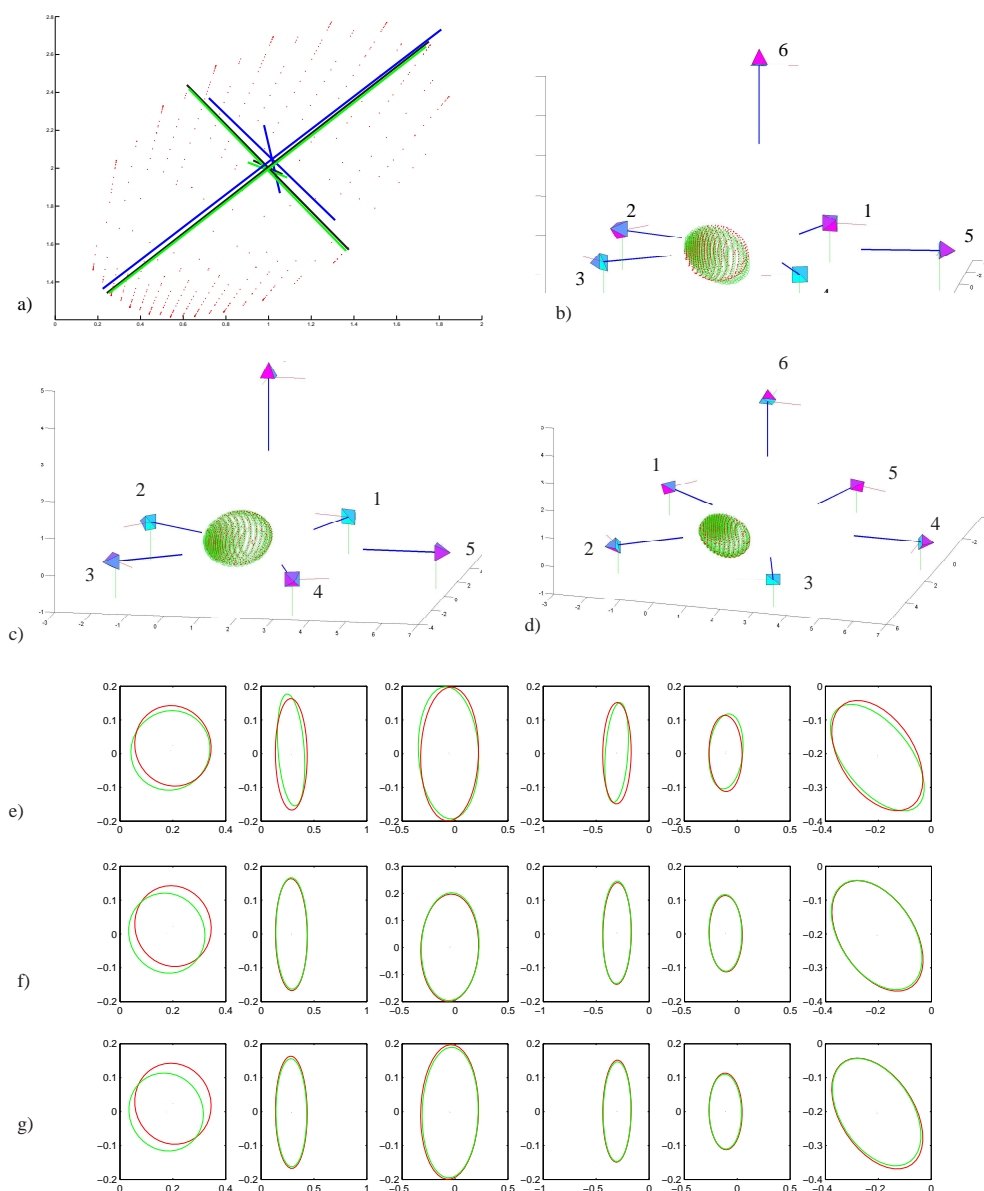


FIG. 7.9 – a) caractéristiques de l'objet : en bleu le résultat de la méthode duale, en noir celui de la méthode non linéaire et en vert celui de la méthode robuste b) reconstruction duale c) reconstruction non linéaire d) reconstruction robuste. Les trois figures horizontales sont les projections de la quadrique reconstruite par la méthode e) duale, f) non linéaire g) robuste

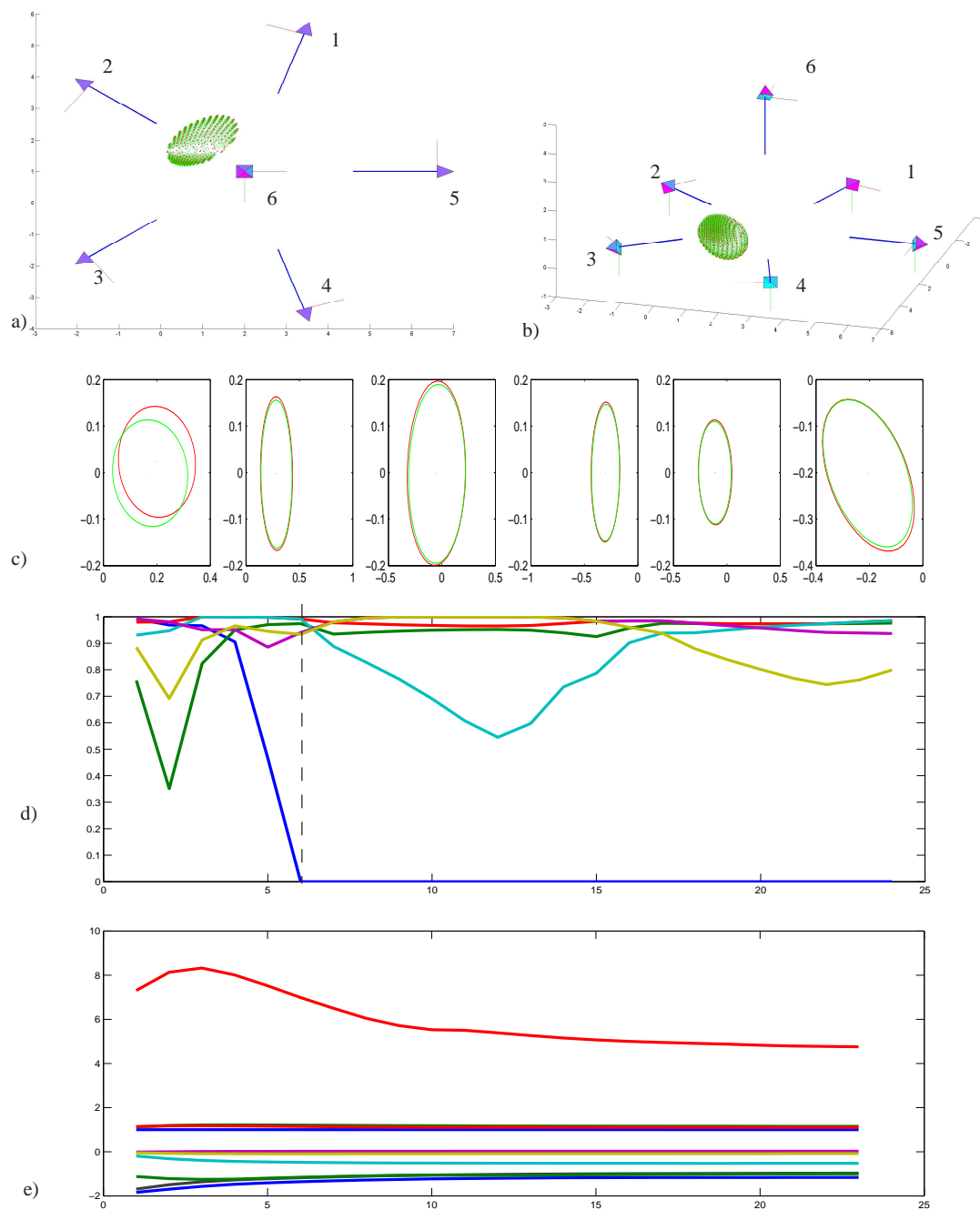


FIG. 7.10 – Simulation faiblement bruitée, reconstruction par la méthode robuste. a) vue de dessus de la quadrique estimée b) vue de cote de la vue estimée c) coniques projetées en rouge et coniques estimées en vert. d) évolution des poids associés à chaque vue e) évolution des paramètres. Après 10 itérations, le poids associé à la première vue chute à zéro.

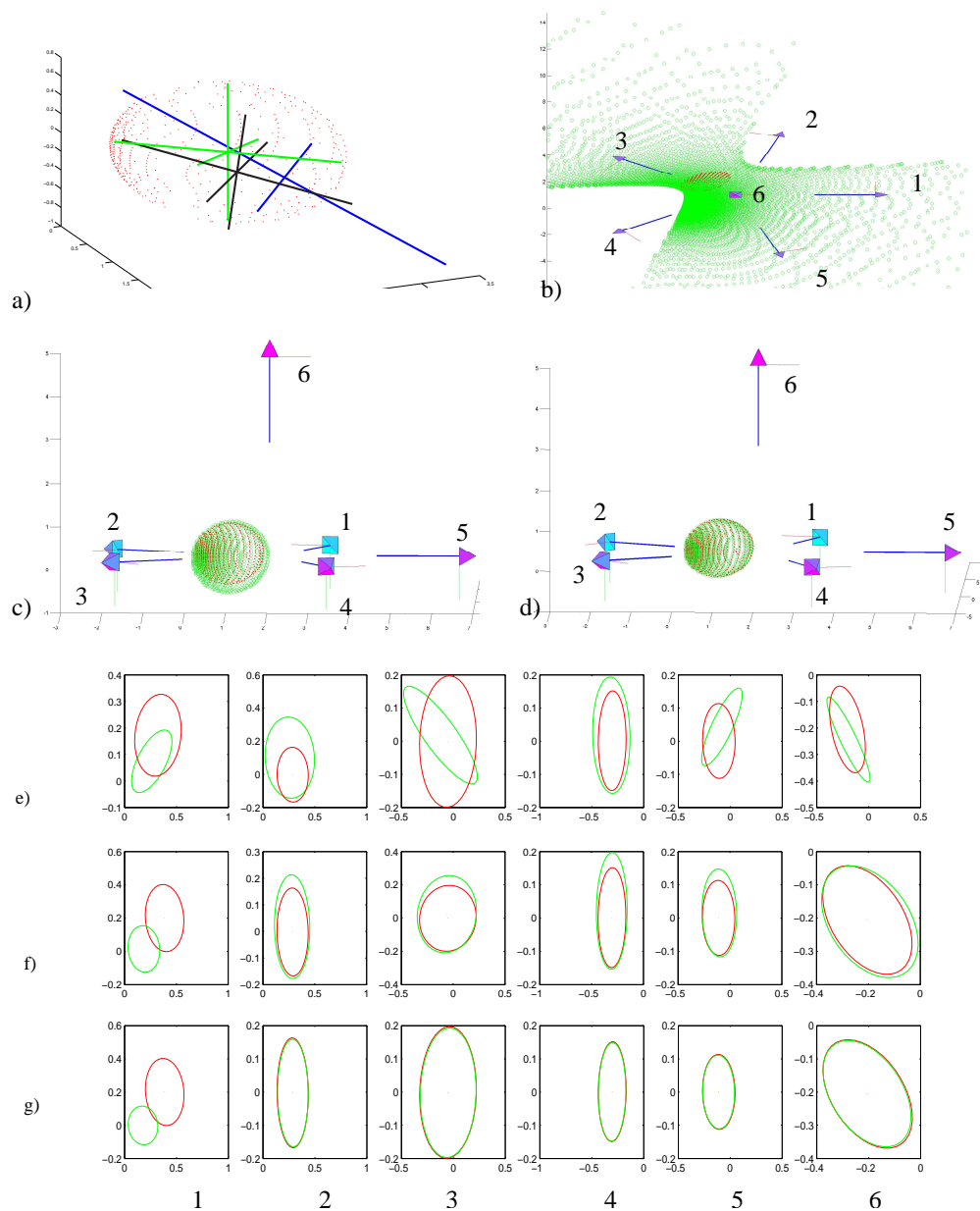


FIG. 7.11 – a) Caractérisations obtenues : en bleu par la méthode duale, en noir par la méthode non linéaire et en vert par la méthode robuste. b) reconstruction duale c) reconstruction non linéaire. d) reconstruction robuste. Les trois figures horizontales sont les projections de la quadrique reconstruite par la méthode e) duale, f) non linéaire g) robuste, dans chacune des vues.

paramètres de la quadrique utilise donc les cinq autres vues. La méthode non linéaire converge également vers un ellipsoïde mais l'estimation des paramètres obtenus est entachée par la contribution de la première vue. Ainsi, la norme infinie de l'erreur entre les paramètres estimée \mathcal{Q}_{nl} et les paramètres réels \mathcal{Q}_r est 1000 fois plus importante que celle entre \mathcal{Q}_{mest} et \mathcal{Q}_r , cette différence étant uniquement due à la contribution de la première vue. Cet écart dans les erreurs se retrouve en calculant la somme des résidus pour toutes les vues exceptée la première. A nouveau, la méthode robuste obtient un score nettement meilleur (voir le tableau 7.3).

	Dimensions			Centre			Orientation			résidu	erreur sur \mathcal{Q}	nombre d'it.
	d_i	d_j	d_k	X_c	Y_c	Z_c	θ_i	θ_j	θ_k			
\mathcal{Q}_r	1	0,71	0,58	1	2	0	80	40	10			
\mathcal{Q}_d	0.46	1.60	-0.9i	1.23	2.30	-0,30	43,00	43,00	95,00	20,10	0.60	1
\mathcal{Q}_{nl}	0.94	0.65	0.55	1.11	2.12	-0.20	90.00	42.28	3.30	$3,6 \cdot 10^{-2}$ ($1 \cdot 10^{-2}$)	$1,1 \cdot 10^{-2}$	10
\mathcal{Q}_{mest}	0.98	0.69	0.57	0.98	1.99	0.00	80.07	40.12	9.93	$6,4 \cdot 10^{-2}$ ($8 \cdot 10^{-5}$)	$1.8 \cdot 10^{-5}$	21

TAB. 7.3 – Comparaison des méthodes de reconstruction dans un cas bruité : un bruit gaussien de variance 0.01 perturbe les positions des points de contours dans la première vue. Les valeurs entre parenthèses correspondent aux résidus calculés sans la première vue.

7.5.3 Reconstruction d'une quadrique à partir de vues proches

Dans cet exemple, la quadrique \mathcal{Q}_r est observée depuis six vues proches les unes des autres. La figure 7.13 présente cet exemple. Dans un premier temps, les projections sont idéales (voir le tableau 7.4 et la figure 7.14), dans un second temps, un bruit vient perturber une des coniques estimées (voir le tableau 7.5 et la figure 7.15).

	Dimensions			Centre			Orientation			résidu	erreur sur \mathcal{Q}	nombre d'it.
	d_i	d_j	d_k	X_c	Y_c	Z_c	θ_i	θ_j	θ_k			
\mathcal{Q}_r	1	0,71	0,58	1	2	0	80	40	10			
\mathcal{Q}_d	1.00	0.70	0.58	1,00	2,00	0,00	80.14	40.03	9.95	$1,16 \cdot 10^{-5}$	$2,6 \cdot 10^{-5}$	1
\mathcal{Q}_{nl}	0.95	0.69	0.56	1.01	1,99	0,00	80.29	42.26	10.57	$1,3 \cdot 10^{-4}$	$1,6 \cdot 10^{-3}$	12
\mathcal{Q}_{mest}	0.98	0.70	0.57	1,00	2,00	0,00	79.87	40.63	10.28	$4,0 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	22

TAB. 7.4 – Comparaison des méthodes pour des estimations non bruitées et six vues proches.

Lorsque les vues sont proches, les reconstructions sont très sensibles au bruit de mesure. Si le bruit est important, toutes les méthodes divergent. Dans cet exemple, le bruit est suffisamment faible pour que toutes les méthodes convergent et les résultats obtenus conduisent à des conclusions identiques à celles déduites des expériences bruitées précédentes : la méthode duale permet d'obtenir un résultat approximatif en un seul pas de calcul et la méthode robuste se montre à nouveau intéressante dans le cas où une des vues est bruitée.

7.5.4 Reconstruction d'un objet de forme quelconque

Dans le premier exemple, l'objet à reconstruire est un parallélépipède rectangle de cotés 1, 2 et 3, de position (2, 2, 1) et d'orientation (0, 45, 0) (voir la figure 7.16). L'objet est observé depuis six vues réparties sur une sphère de vision. Les résultats de reconstruction obtenus sont résumés dans le tableau 7.6 et par la figure 7.17.

La méthode duale permet d'obtenir un résultat très satisfaisant. Les meilleurs résultats sont obtenus lorsque les méthodes non linéaires sont initialisées par le résultat de la méthode duale. Ces

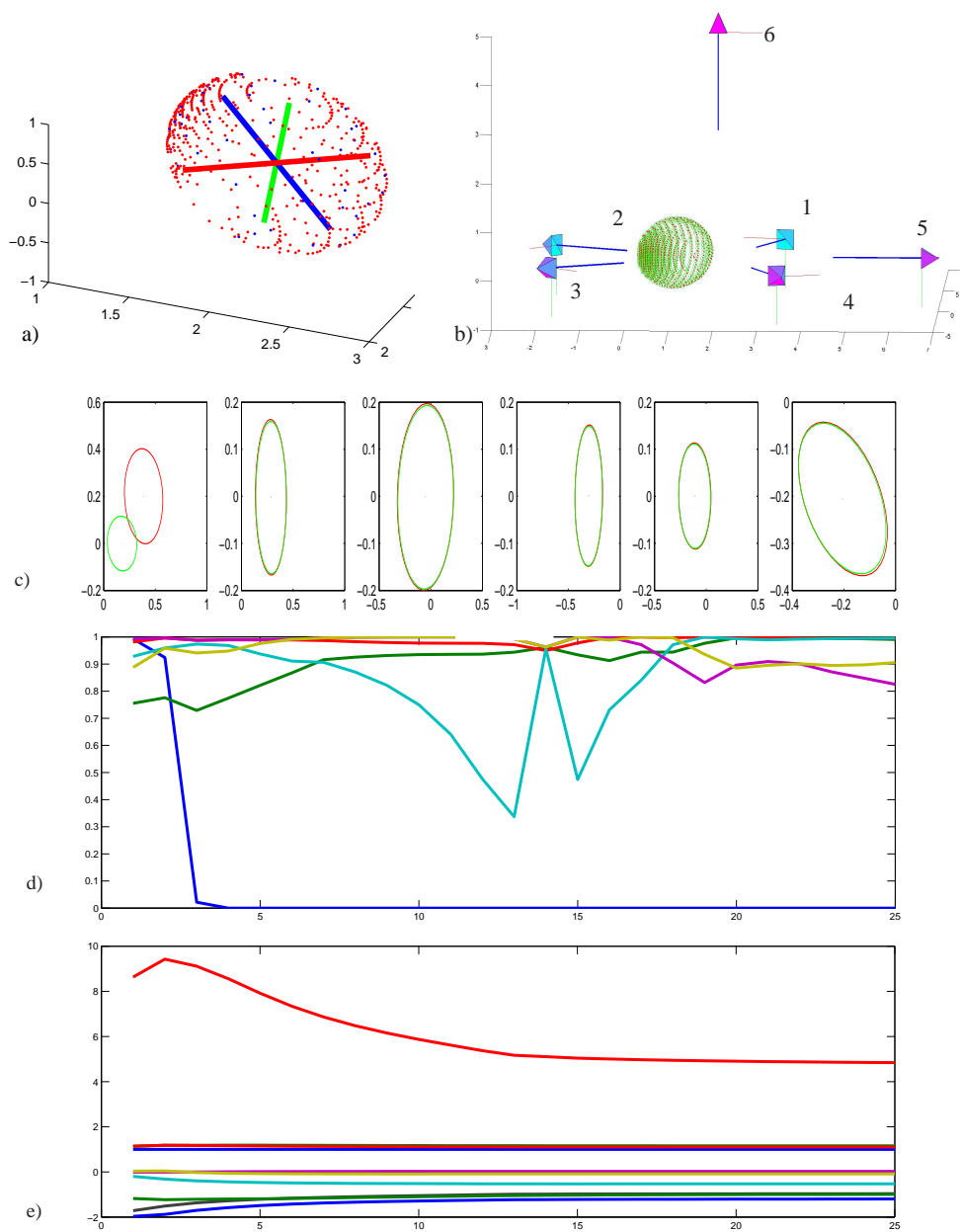


FIG. 7.12 – Simulation fortement bruitée, reconstruction par la méthode robuste. a) Caractéristiques déterminées par la méthode robuste b) vue de côté de la quadrique estimée c) estimation des coniques projetées. d) Évolution des poids. e) Évolution des paramètres.

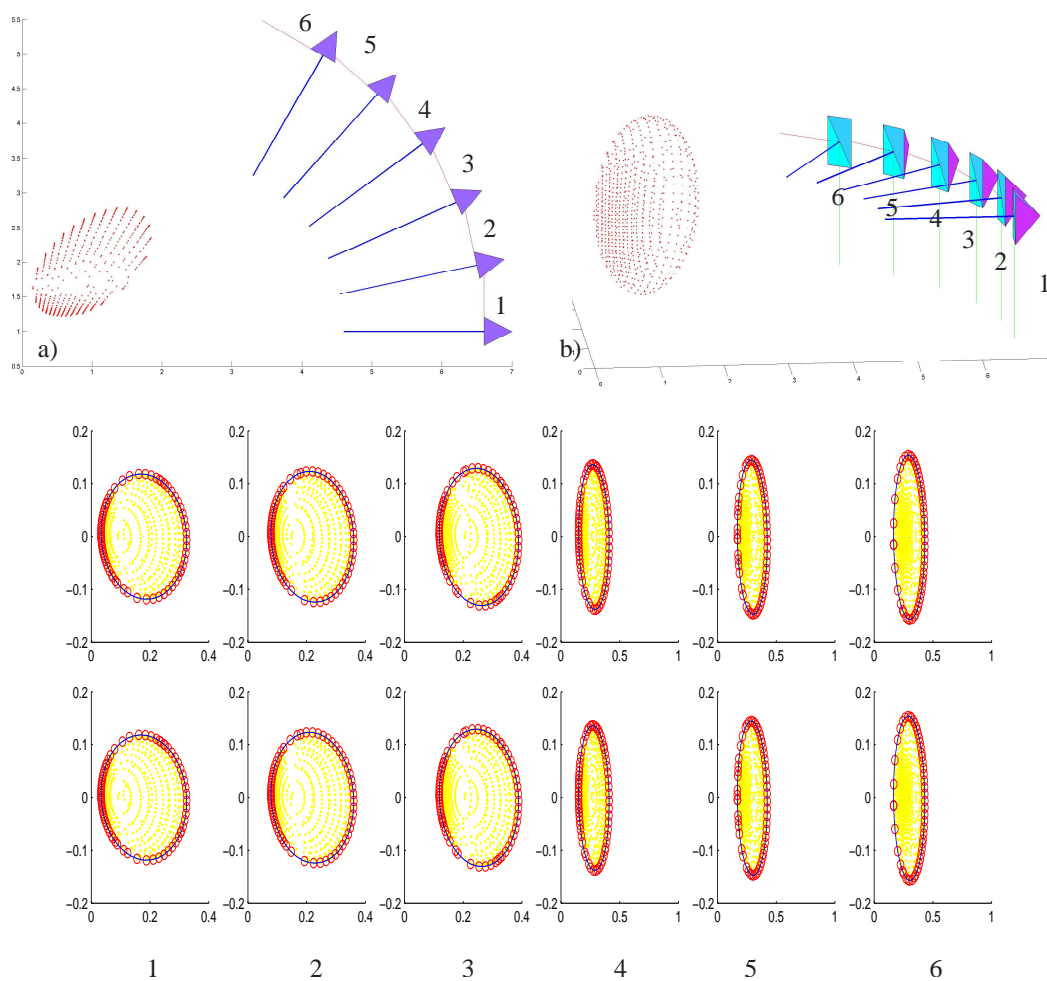


FIG. 7.13 – Vues proches : une quadrique \mathcal{Q}_r (en rouge) est observée depuis six points de vues (pyramides) proches. a) vue du dessus, b) vue de côté c) projections non bruitée dans les plans images d) projection bruitée dans les plans images. Les points jaunes sont les projections des points de l'objet, les cercles rouges sont les points de contours détecté et les ellipses bleues sont les ellipses estimées.

	Dimensions			Centre			Orientation			résidu	erreur sur \mathcal{Q}	nombre d'it.
	d_i	d_j	d_k	X_c	Y_c	Z_c	θ_i	θ_j	θ_k			
\mathcal{Q}_r	1	0,71	0,58	1	2	0	80	40	10			
\mathcal{Q}_d	0,86	0,66	0,49	1,11	2,04	0,00	66.59	63.05	3.63	$1,4 \cdot 10^{-2}$ ($8,7 \cdot 10^{-3}$)	$1,6 \cdot 10^{-1}$	1
\mathcal{Q}_{nl}	0,87	0,63	0,51	1,00	1,97	0,00	80.25	42.16	10.75	$1,1 \cdot 10^{-3}$ ($2,4 \cdot 10^{-3}$)	$1,9 \cdot 10^{-3}$	8
\mathcal{Q}_{mest}	0,97	0,69	0,57	1,00	1,99	0,00	79,75	40,79	10,39	$3,5 \cdot 10^{-4}$ ($7,5 \cdot 10^{-5}$)	$2,5 \cdot 10^{-4}$	16

TAB. 7.5 – Comparaison des méthodes de reconstruction dans un cas où l'objet est observé depuis des vues proche et pour lequel une des coniques estimées est bruitée (variance 0.005 et moyenne 0.01).

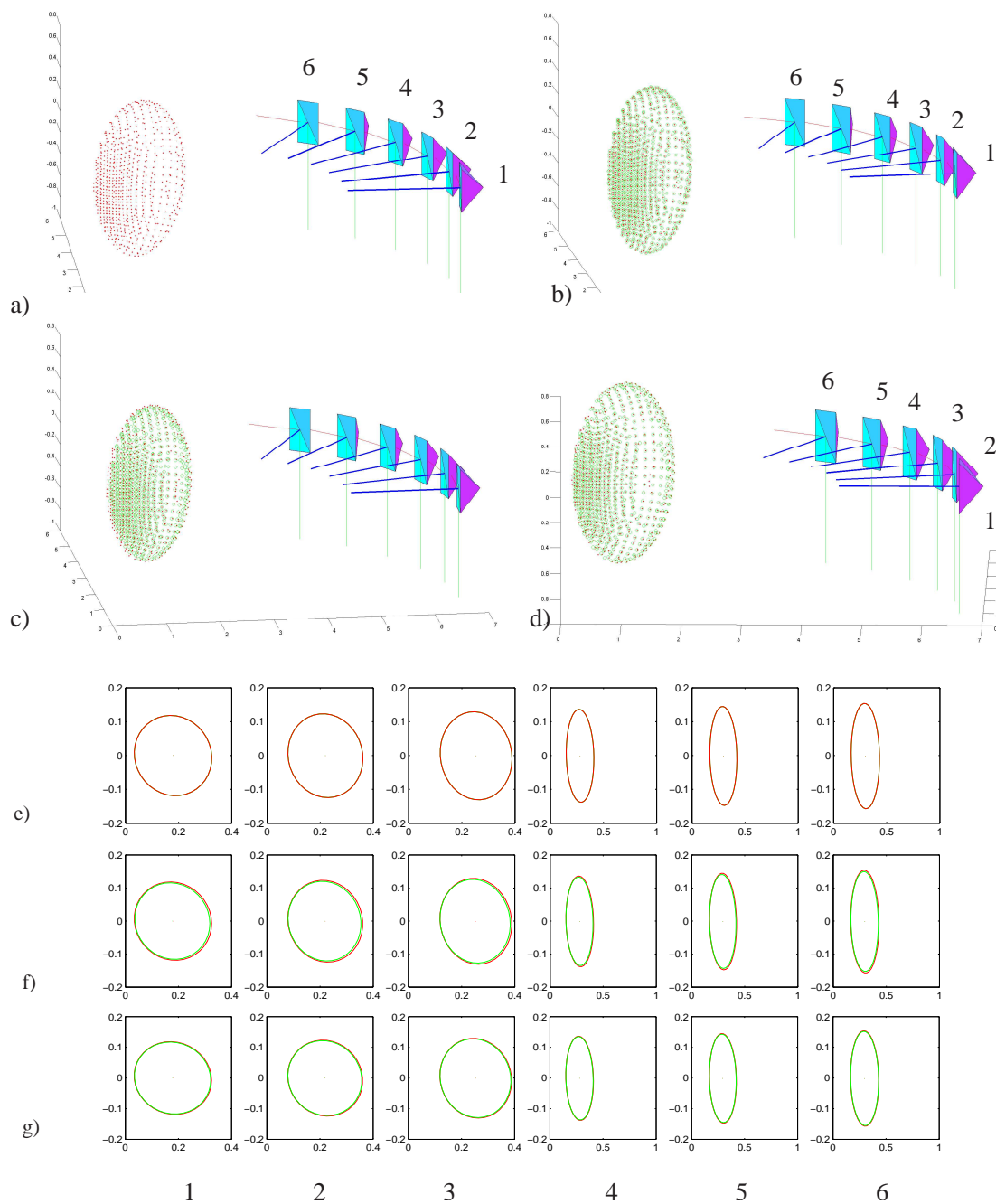


FIG. 7.14 – a) objet réel b) reconstruction duale c) reconstruction non linéaire. d) reconstruction robuste. Les trois figures horizontales sont les projections de la quadrique reconstruite par la méthode e) duale, f) non linéaire g) robuste

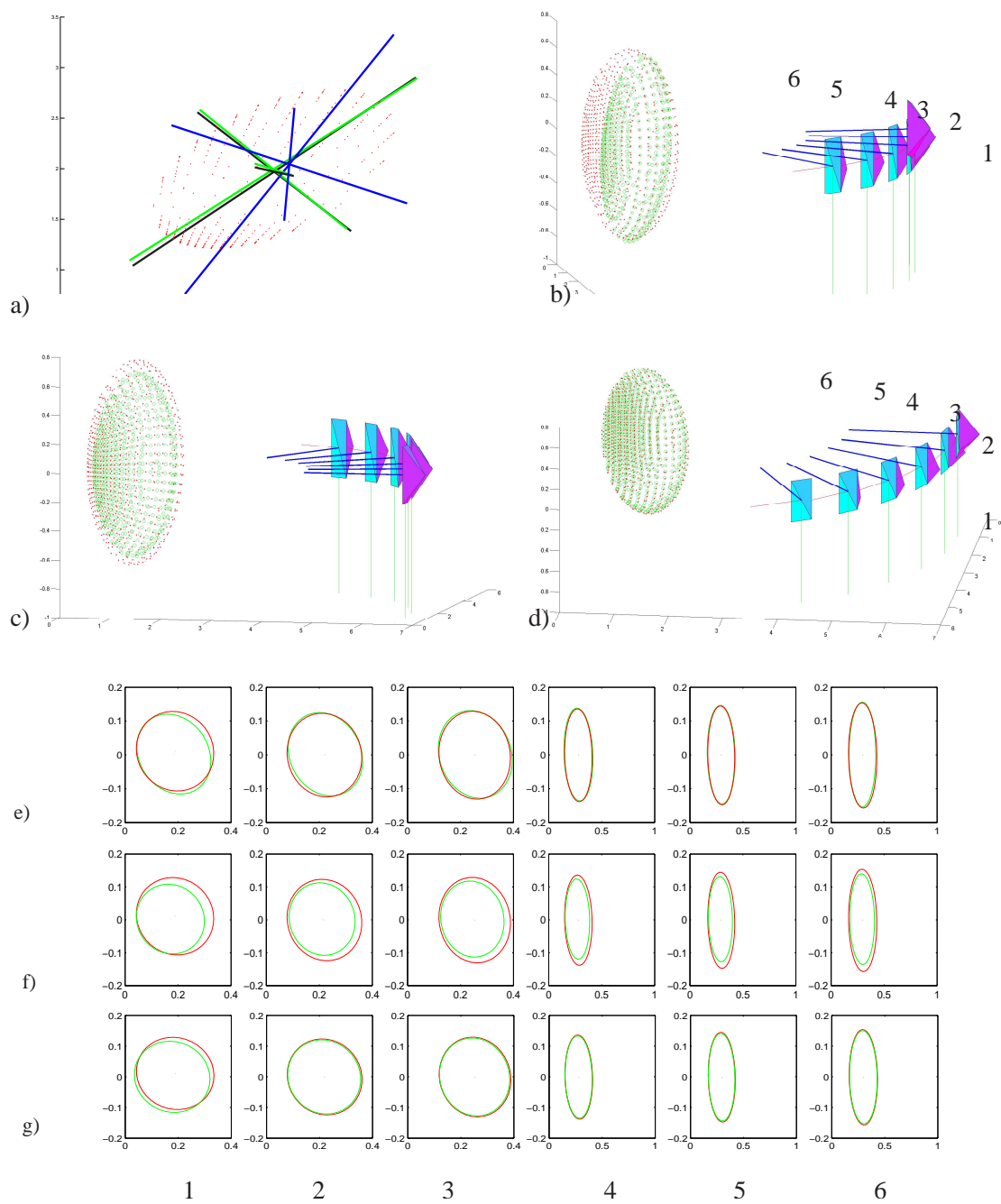


FIG. 7.15 – a) objet réel b) reconstruction duale c) reconstruction non linéaire, d) reconstruction robuste. Les trois figures horizontales sont les projections de la quadrique reconstruite par la méthode e) duale, f) non linéaire g) robuste, dans chaque vue.

méthodes sont respectivement notées \mathcal{Q}_{dnl} \mathcal{Q}_{dmest} et les résultats obtenus sont consignés dans le tableau récapitulatif. Les résultats de l'estimation de l'orientation ne sont pas identiques à l'orientation réelle mais permettent cependant d'obtenir un ellipsoïde bien orienté. Ceci est dû au fait que l'orientation peut être déterminée de plusieurs manières à cause des symétries de l'ellipsoïde.

Un second exemple est illustré sur la figure 7.18. Il s'agit d'un objet de forme quelconque composé d'un parallélépipède et d'ellipsoïdes. Les trois méthodes permettent d'estimer la forme de la quadrique qui correspond aux coniques estimées dans les vues. Cependant, pour cet objet les axes obtenus ne correspondent pas tout à fait aux axes de préhension naturelle. Les deux petits ellipsoïdes sont légèrement décalés à l'arrière de l'axe vertical de l'ellipsoïde le plus grand et du cube. C'est pourquoi la quadrique reconstruite est légèrement inclinée.

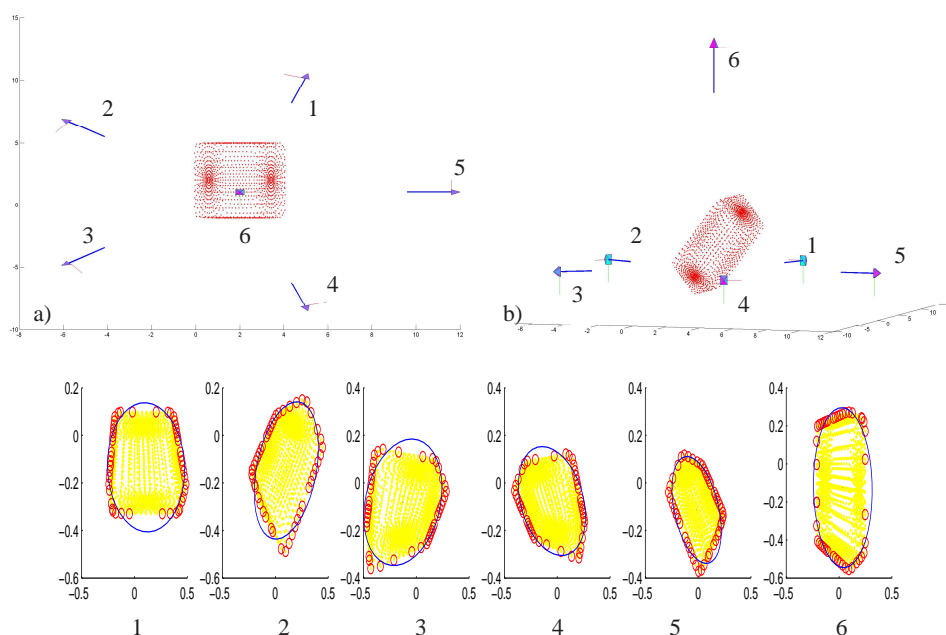


FIG. 7.16 – Un parallélépipède est observé depuis six vues. a) vue de dessus b) vue de côté c) projections dans les six vues

méthode	Dimensions			Centre			Orientation			résidu	nombre d'it.
	d_i	d_j	d_k	X_c	Y_c	Z_c	θ_i	θ_j	θ_k		
parallélépipède	3	2	1	2	2	1	0	45	0		
\mathcal{Q}_d	3,64	2,95	1,48	1,83	2,00	1,00	-0,16	-47,66	89,17	$2 \cdot 10^{-2}$	1
\mathcal{Q}_{nl}	3,70	3,08	1,80	1,76	1,95	1,03	0,93	-45,27	-88,18	$5 \cdot 10^{-3}$	20
\mathcal{Q}_{mest}	3,72	3,09	1,78	1,62	1,91	1,00	1,74	135,35	89,35	$5 \cdot 10^{-3}$	37
\mathcal{Q}_{dnl}	3,75	3,10	1,79	1,81	1,99	1,03	0,96	-45,48	-88,14	$6 \cdot 10^{-3}$	25
\mathcal{Q}_{dmest}	3,65	3,04	1,58	1,99	2,00	1,00	-47,61	-47,61	-85,45	$1 \cdot 10^{-3}$	33

TAB. 7.6 – Comparaison des méthodes pour la reconstruction d'un objet de forme quelconque : un parallélépipède de dimensions 1, 2 et 3, de position (2, 2, 1) et d'orientation (0, 45, 0)

7.6 Conclusion

Ces premières observations permettent de dresser un premier bilan quant à la méthode de reconstruction la plus appropriée à notre application :

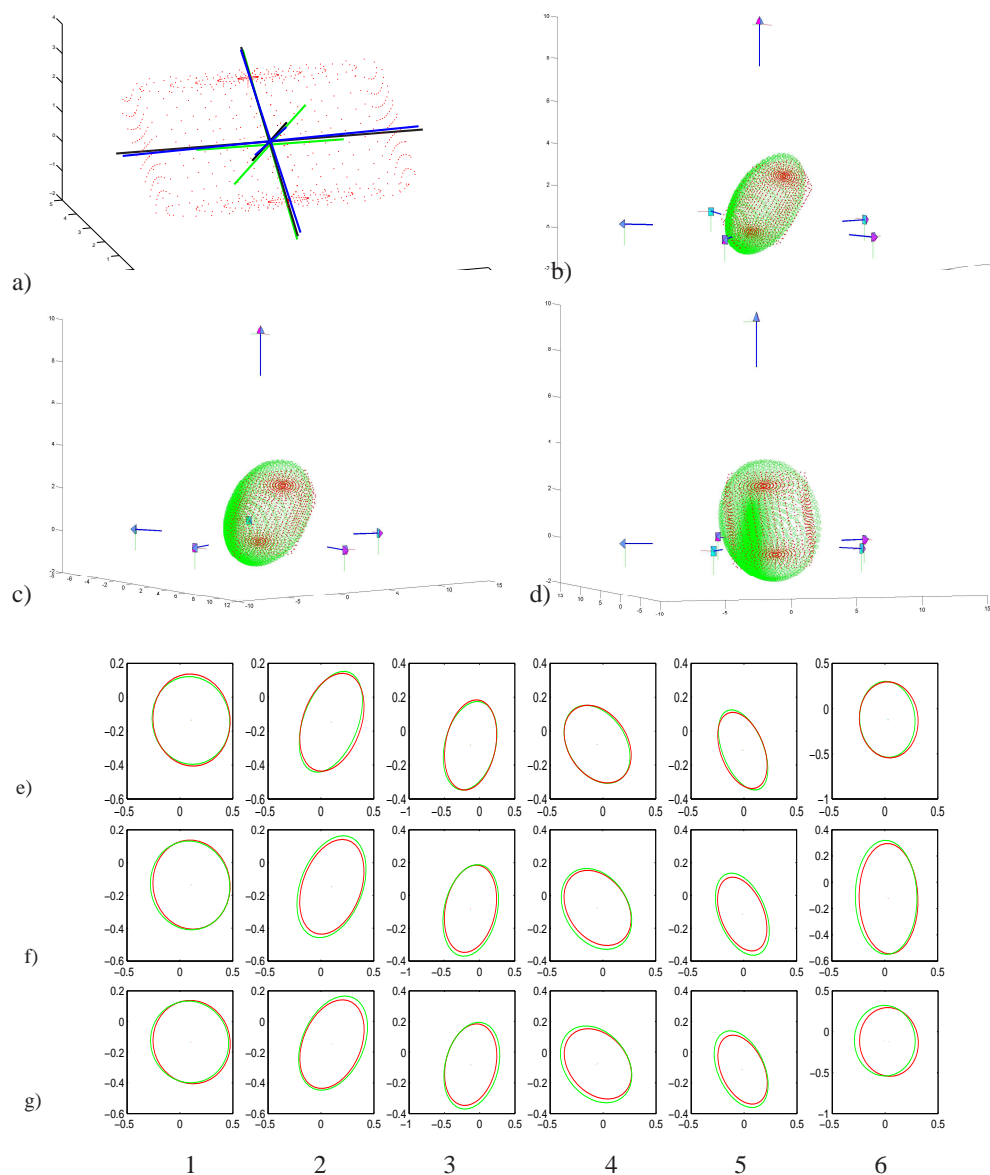


FIG. 7.17 – Estimation de la quadrique qui correspond au mieux à un objet de forme parallélépipédique qui est observé depuis six vues. a) Les caractéristiques estimées par les différentes méthodes : en bleu la méthode duale, en noir la méthode non linéaire, en vert la méthode robuste. Reconstructions obtenues : b) par la méthode duale, c) par la méthode non linéaire c) par la méthode robuste. Projections associées à d) la méthode duale e) la méthode non linéaire f) la méthode robuste.

- **Sensibilité au bruit de mesure :**
les méthodes proposées permettent de retrouver les axes principaux d'un objet avec une assez bonne précision même lorsque les données sont bruitées.
- **Nombre de caméras :**
plus le nombre de caméras est important, meilleure est la reconstruction.
- **Positionnement des caméras :**
la position des points de vue joue un rôle essentiel dans la reconstruction. Si les points de vue sont trop proches, les paramètres de la quadrique seront mal estimés. À l'inverse, un choix judicieux

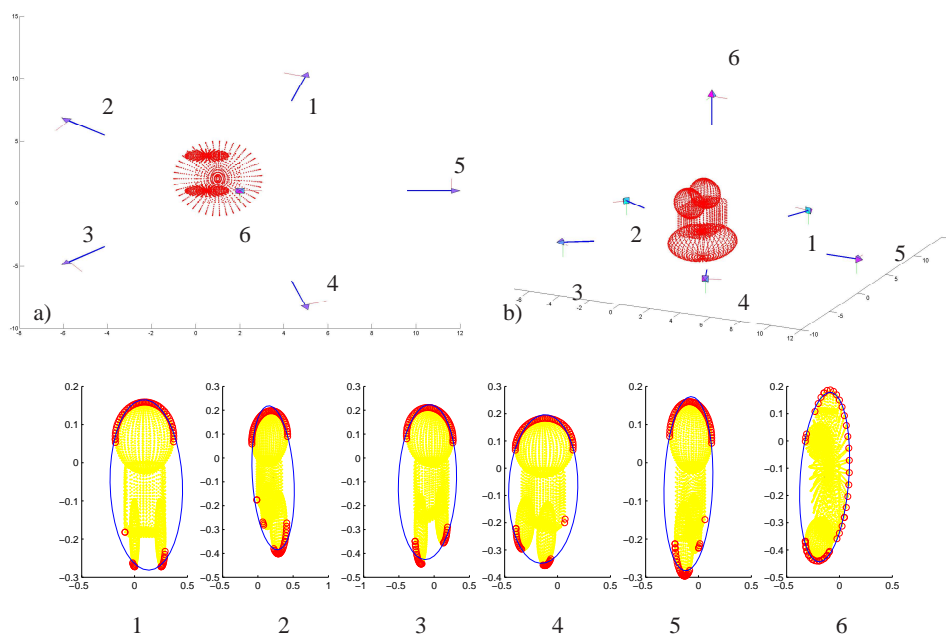


FIG. 7.18 – Un objet de forme quelconque est observé depuis six vues. a) vue de dessus b) vue de côté c) projections dans les six vues

des points de vues permet d'améliorer l'estimation des paramètres.

- **Généricité de la caractérisation** : le calcul des axes principaux et des dimensions principales peut être effectué aussi bien sur des objets dont la forme épouse parfaitement celle d'une quadrique que sur des objets de forme quelconque, dès lors qu'ils sont globalement convexes.

Les méthodes proposées permettent de caractériser la forme d'un objet en calculant ses axes principaux et ses dimensions selon ces axes. Elle est valide tant que l'objet considéré est globalement convexe. Dans la suite, la méthode duale sera utilisée pour initialiser le système de résolution non linéaire robuste en prenant soin de vérifier que son résultat est une quadrique régulière. Dans le cas contraire la méthode non linéaire sera initialisée par une sphère de rayon arbitraire (20cm) et centrée sur la position estimée.

Dans le chapitre à venir, nous présenterons une méthode de traitement d'image qui permet de détecter les contours de l'objet dans les vues tout en proposant des pistes vers une initialisation de ces méthodes à partir d'un seul *click*. L'avant dernier chapitre propose une méthode de vision active permettant de sélectionner les vues optimales pour la reconstruction. Enfin un dernier chapitre présente les résultats de reconstruction obtenus pour des objets réels.

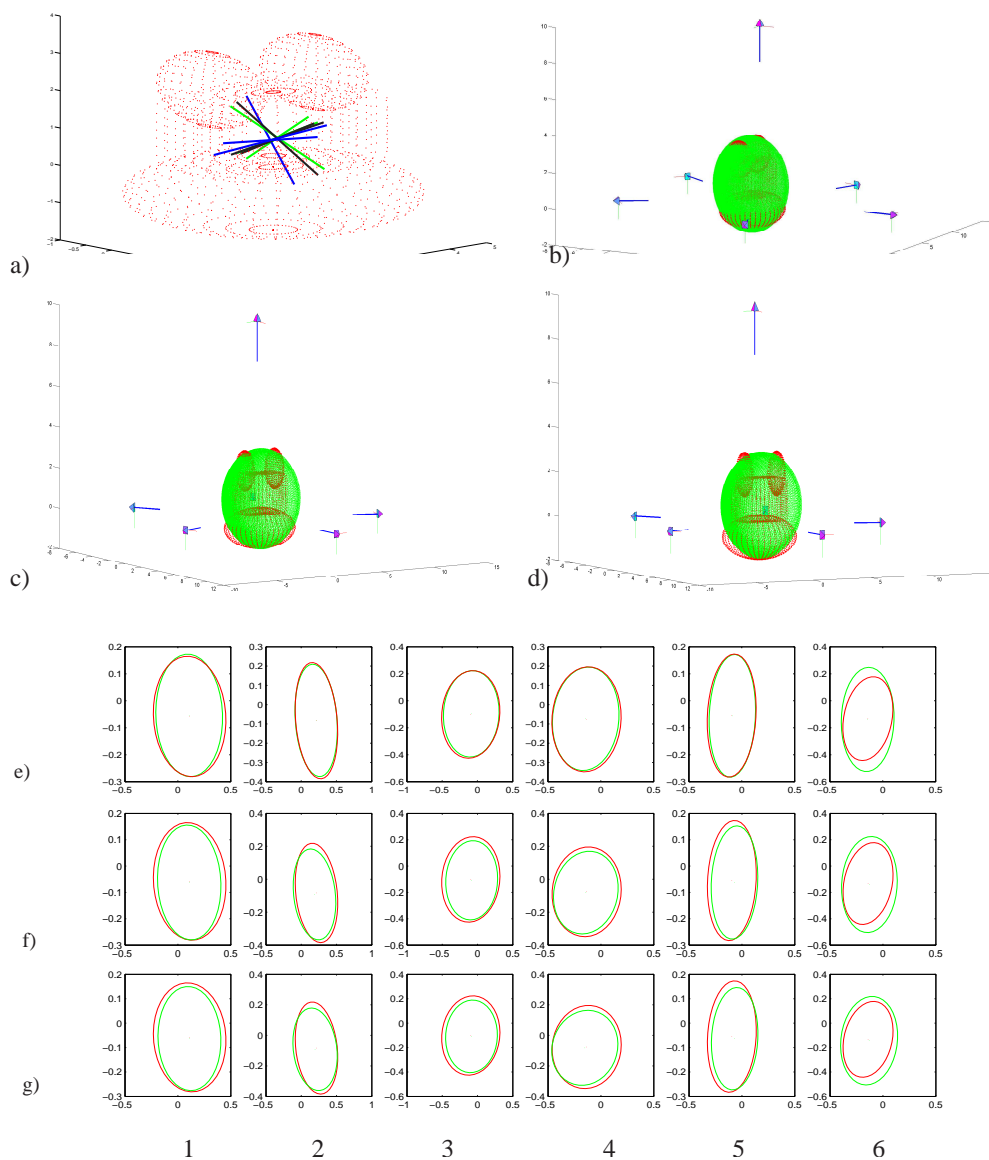


FIG. 7.19 – Objet de forme quelconque a) ses caractéristiques estimées par les différentes méthodes : en bleu la méthode duale, en noir la méthode non linéaire, en vert la méthode robuste. Reconstructions obtenues : b) par la méthode duale, c) par la méthode non linéaire d) par la méthode robuste. Projections associées à e) la méthode duale f) la méthode non linéaire g) la méthode robuste.

Détection de l'objet par une méthode de contours actifs

La segmentation d'un objet inconnu dans une image est un problème ouvert. Bon nombre de travaux de traitement de l'image portent sur cette thématique, cependant il n'existe pas à notre connaissance de solution universelle à ce problème, c'est à dire une méthode qui permette de détecter tout type d'objet placé sur un arrière-plan quelconque. Dans la première partie de cette section, un bref état de l'art présente les méthodes utilisées dans les applications d'assistance robotisée à la saisie.

Parmi les méthodes existantes, les méthodes basées sur une base de données d'objets donnent de bon résultats de détection sans contrôle de l'illumination. La mise en correspondance de la vue courante avec les informations stockées permet d'identifier l'objet et d'en estimer la pose. Les contours de l'objet peuvent ainsi être directement projetés à partir des informations connues *a priori*, par exemple en utilisant une mise en correspondance de points SIFT comme le propose D.G. Lowe [Lowe 04]. Dans le cadre de cette étude, nous avons choisi de ne pas utiliser de base de données afin de ne pas limiter la saisie à un ensemble d'objets connus *a priori* (voir Hypothèse 3).

Dans la seconde section de ce chapitre une méthode de détection des contours reposant sur des contours actifs sera proposée. Cette méthode permet de faire évoluer une estimation grossière des contours vers les contours de plus fort contraste. Nous supposons ici que ces contours de plus fort contraste sont ceux de l'objet réel.

Hypothèse 13 *Le contour de plus fort contraste est le contour de l'objet.*

De plus, l'objet sera supposé entièrement inclus dans chacune des images de la séquence servant à la reconstruction.

Hypothèse 14 *L'objet est entièrement compris dans chacune des vues.*

8.1 Méthodes existantes.

La plupart des outils de saisie qui utilisent la vision sont basés sur des méthodes pragmatiques permettant la segmentation de l'objet à saisir sur le fond. Bien souvent, le traitement de l'image est simplifié par une série d'hypothèses. Et l'idée est couramment admise que les solutions de traitement

pour des scènes complexes viendront des recherches menées par les spécialistes en traitement de l'image.

Selon les hypothèses sur l'objet et la scène, plusieurs solutions ont ainsi été proposées pour extraire les contours d'un objet dans une image. Par exemple, si l'objet est de couleur relativement uniforme, ou bien si la couleur du fond est relativement uniforme, il est possible d'utiliser des techniques basées sur une segmentation des images par détection des régions d'une même couleur pour délimiter le contour de l'objet [Sanz 05, Tsui 08, Glover 08]. Une telle segmentation des objets est illustrée sur la figure 8.1 [Glover 08]. Dans cette étude, les objets présentés sont de couleur uniforme et distincte.

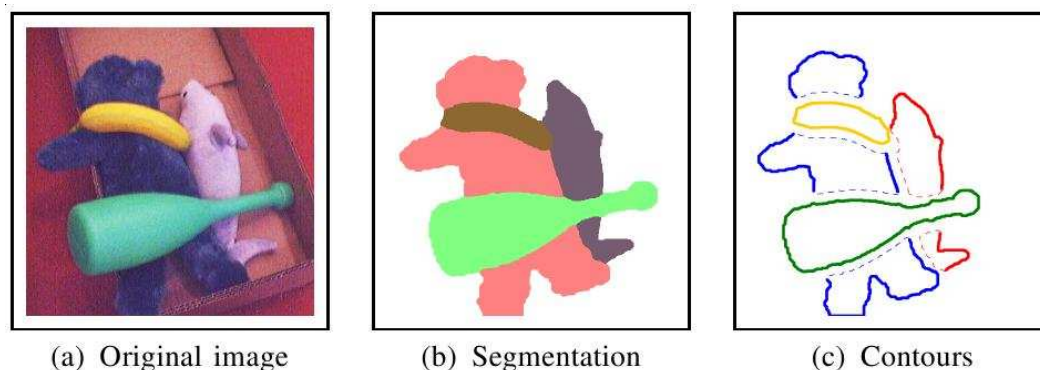


FIG. 8.1 – Segmentation à partir de la couleur d'un objet et détection des contours associée. Les objets sont de couleurs uniformes et chaque objet a une couleur discriminante [Glover 08].

À l'inverse, si l'objet n'est pas uniforme mais texturé, il peut être supposé que la zone de l'image contenant la plus forte densité de contours contient l'objet [Becker 98]. Dans un premier temps, les contours sont détectés en utilisant une méthode robuste aux changements d'échelle et reposant sur les ondelettes [Mallat 99]. Les zones de plus forte densité de contours sont ensuite repérées et représentées par des régions d'intérêt, ou *blob*. Dans le projet Gripsee [Becker 98], le *blob* sélectionné est celui qui est pointé par la main de l'utilisateur (voir figure 8.2). Dans notre cas, le *blob* contenant l'objet serait celui qui contiendrait le *clik*.

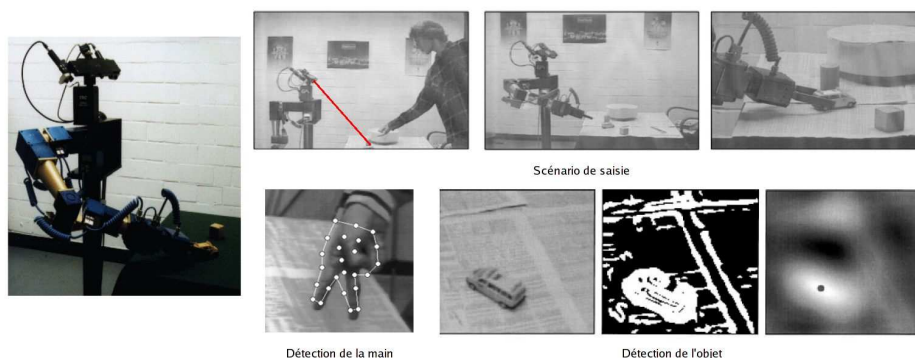


FIG. 8.2 – Comme dans le cadre de notre étude, l'objet est désigné par pointage. La segmentation repose sur l'hypothèse que l'objet se trouve à l'emplacement de la zone la plus texturée de l'image. À gauche l'image à traiter, au centre les contours détectés et à droite la détection du *blob* contenant l'objet.

Plutôt que d'imposer des hypothèses sur la texture des objets, [Li 08] propose une méthode de segmentation très prometteuse qui suppose que les objets présentent un axe de symétrie, conduisant ainsi à la détection des contours symétriques dans une image. Une telle approche permet de localiser

en temps réel la plupart des objets du quotidien, même s'ils sont transparents. La figure 8.3 présente quelques résultats de segmentation obtenus avec cet algorithme.

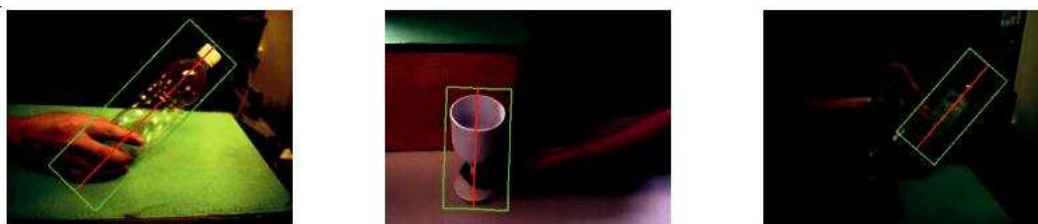


FIG. 8.3 – Segmentation par recherche des symétries [Li 08].

Dans le cadre d'une approche multi-images, où plusieurs vues calibrées sont disponibles, il est possible de différencier l'objet du fond en introduisant des informations sur la structure de la scène : calcul de carte de profondeur, détection de plans, construction de nuages de points denses, représentation volumique, etc. En parallèle de notre travail, une étude est menée actuellement au CEA pour segmenter l'objet à partir d'une fusion d'informations de disparité, obtenue grâce à un banc de stéréovision, et d'informations couleurs. L'idée est de segmenter l'image en régions de couleurs puis de fusionner les zones qui ont la même disparité (voir la figure 8.4). D'autres travaux se sont portés sur de la segmentation mixte, couleur et disparité, par exemple [Benamri 04, Shibuya 08]. Dans [Benamri 04], la distance est représentée par un ensemble de portions planes. Pour lever l'hypothèse d'une scène texturée, il est possible d'utiliser un capteur laser qui permet d'avoir une information de distance à l'obstacle, comme proposé dans [Shibuya 08].



FIG. 8.4 – Segmentation reposant sur une carte de disparité et la détection de région d'une même couleur [LeCam 08].

Si la scène est texturée et si la caméra est en mouvement, une segmentation au sens du mouvement peut être employée [Odobez 95]. Les différents objets composant la scène sont identifiés en recherchant les zones qui ont un mouvement cohérent. Ces méthodes ont été utilisées dans le cadre de l'exploration de scène [Flandin 00]. Les objets sont placés sur un support plan et texturé, ils peuvent être identifiés en détectant les points dont la trajectoire exclut l'appartenance au plan.

Les vues calibrées d'une scène peuvent être utilisées pour obtenir un nuage de points denses par mise en correspondance de points d'intérêt (voir paragraphe 4.3.1.1) [ElKhoury 07]. Les objets sont alors différenciés du fond par segmentation de l'information 3D en s'intéressant à la structure d'un nuage de points ou bien en décomposant la scène en un ensemble de fonctions implicites, par exemple des super-ellipsoïdes [Whaite 97b, Whaite 97a].

Par ailleurs, si un ensemble de vues étalonnées est disponible, une représentation *voxellique* des objets peut être obtenue par des méthodes de *voxel coloring* ou *space carving*. Ces méthodes reposent sur la notion de *photo-consistance* des points d'une scène pour laquelle la direction d'illumination est constante, c'est-à-dire qu'un point de la surface d'un objet sera vu avec une même couleur quelle que soit la position de l'observateur. Ces méthodes ont été testées durant le stage de Master de Caroline Nadeau [Nadeau 08, Nadeau 09].

Dans le cadre de cette étude, nous proposerons une méthode basée sur les techniques de contours actifs [Kass 87, Terzopoulos 91] pour détecter le contour des objets qui présente le plus fort contraste.

Les contours actifs, ou modèles déformables, ont déjà été utilisés avec succès dans des travaux de saisie d'objets guidée par la vision [Blake 95] et de modélisation [Chung 96]. Ils présentent un moyen efficace d'obtenir un contour fermé et permettent le cas échéant le suivi dans une séquence d'images du contour détecté.

Un inconvénient de ces méthodes est qu'elles demandent une étape d'initialisation au cours de laquelle un contour approximatif de l'objet est défini. En particulier, lorsque l'objet est placé sur un fond texturé, ce contour initial doit être suffisamment proche du contour réel pour que l'algorithme converge. Dans le cadre de notre application, un unique *clic* est demandé à l'utilisateur. Une seconde section présentera quelques pistes de recherche qui pourraient mener à une initialisation du contour actif en un seul *clic*.

8.2 Détection et suivi de contours par contours actifs

Cette partie présente l'algorithme de contours déformables développé par C. Collewet [Collewet 09] qui a été utilisé dans le cadre de cette thèse [Dune 08a]. Dans un premier temps, nous rappellerons le principe général des contours actifs, puis la méthode développée par C. Collewet et les résultats obtenus sur des images réelles.

8.2.1 Principe des contours actifs

Depuis leur première introduction [Kass 87] les méthodes dites *contours actifs*, *modèles déformables* ou *snakes*, ont prouvé leur efficacité et leur robustesse dans des contextes aussi variés que la segmentation, la détection et le suivi de contour. Ces méthodes sont basées sur l'évolution d'un contour qui peut être fermé, depuis une position initiale vers une position d'équilibre, approchant au mieux le contour réel (voir figure 8.5).

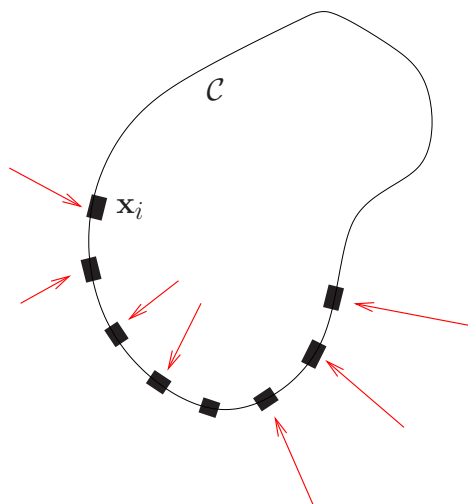


FIG. 8.5 – Contour actif : l'évolution des contours est soumise à des forces externes (dues à l'image) et internes (dues au contour). Dans la version classique des contours actifs [Kass 87], chaque point se déplace en fonction des forces qui lui sont appliquées (flèches rouges).

Les équations d'évolution du contour permettant de l'amener à une position d'équilibre sont obtenues grâce à l'analogie du contour dans l'image et d'un système physique de topologie identique plongé dans un liquide visqueux. La dynamique du contour déformable est alors régie par des forces qui s'exerceraient sur le système physique analogue (voir la figure 8.6). La courbe de contour \mathcal{C} est

assimilée à une infinité de points matériels \mathbf{x}_i (comme illustré sur la figure 8.6). Ces points sont baignés dans un liquide visqueux et sont soumis à un champ de forces extérieures qui dépendent des données présentes dans l'image (le plus souvent en fonction du gradient spatial) et à des forces de liaisons élastiques qui sont en fait des termes de régularisation de Tikhonov [Tikhonov 63]. Ces termes vont tendre à limiter les capacités du contour actif à se déformer, comme à s'allonger ou à modifier sa courbure locale. Ainsi, à chaque itération, chaque point se déplace en fonction des forces auxquelles il est soumis.

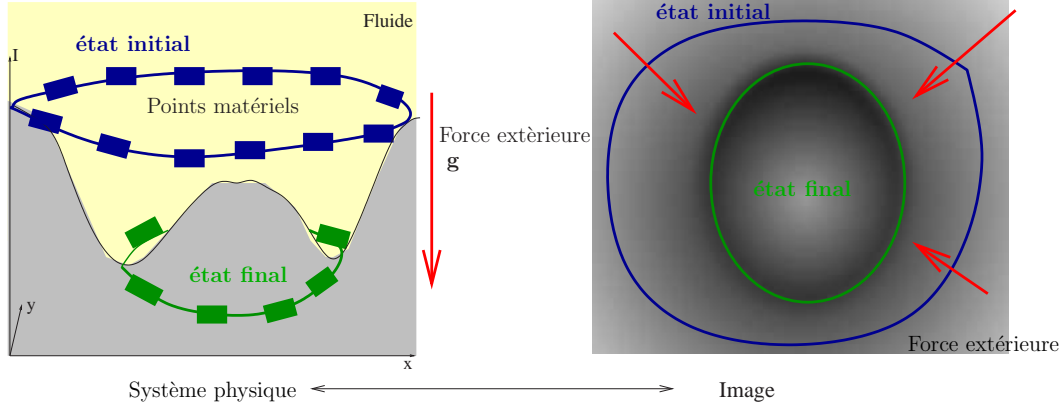


FIG. 8.6 – Analogie contour actif - système physique : le contour peut être assimilé à un système physique de topologie similaire plongé dans un fluide et soumis à des champs forces, par exemple, la gravité et les forces de frottement fluide.

Il existe différents types de contours actifs reposant chacun sur des représentations différentes des contours (ensemble discret de points, approximation polygonale, courbe paramétrée, courbe d'isovaleur 0 d'une image d'un ensemble de niveaux) et des schémas numériques particuliers. Les approches existantes peuvent être classifiées en deux catégories, les *snakes géométriques* [Casselles 97, Malladi 95] pour lesquels les contours sont décrits d'après une représentation d'ensemble de niveaux, ou *levelsets* ; et les *snakes paramétriques* qui sont soit représentés par un ensemble discret de points [Kass 87], soit approximés par des fonctions paramétriques adéquates, telles que les fonctions B-Splines [Menet 90, Bascle 92], ou des superquadriques [Terzopoulos 91]. Comme les approches géométriques sont très coûteuses en temps de calcul [Precioso 05], l'approche proposée est du type paramétrique, afin de répondre au mieux aux contraintes temps réel imposées par notre application.

8.2.2 Méthode développée

En considérant le problème analogue à celui d'un système mécanique élastique soumis à un champ de forces extérieures et plongé dans un liquide visqueux, les équations d'évolution du système peuvent être obtenues grâce au formalisme de Lagrange :

$$\frac{d}{dt} \left\{ \frac{\partial L}{\partial \dot{a}_i} \right\} - \frac{\partial L}{\partial a_i} = \Theta_i(a_i) \quad (8.1)$$

où $L = T - U$ est le Lagrangien du système, U son énergie potentielle, par exemple les termes de Tikhonov, T son énergie cinétique et $\Theta_i(a_i)$ la force généralisée relative à l'ensemble des forces extérieures qui lui sont appliquées et associée à la variable généralisée a_i .

L'approche retenue repose sur une description paramétrique des contours par une variable de parcours $u \in [0, 2\pi]$ et par le vecteur \mathbf{a} constitué des variables généralisées a_i [Menet 90, Terzopoulos 91, Bascle 92].

$$\mathbf{x}_a(u) = \begin{pmatrix} x_a(u) \\ y_a(u) \end{pmatrix} \quad (8.2)$$

Dans ces conditions et en supposant que les points matériels constituant le contour sont de même masse μ , il vient :

$$T = \int_{\mathcal{C}_a} \frac{1}{2} \mu \left\| \frac{\partial \mathcal{C}_a(u)}{\partial t} \right\|^2 du \quad (8.3)$$

$$\Theta = \int_{\mathcal{C}_a} \mathbf{F}_{ext}(u)^\top \frac{\partial \mathcal{C}_a(u)}{\partial a_i} du \quad (8.4)$$

où $\mathbf{F}_{ext}(u)$ est l'ensemble des forces extérieures s'appliquant au point (x_a, y_a) comme les forces dues à des caractéristiques propres de l'image, des forces de frottement fluide $-\gamma \int_{\mathcal{C}_a} \frac{\partial \mathcal{C}_a(u)}{\partial a_i} du$ ou des forces de soufflage ou de pression. Comme proposé dans [Cohen 93], l'énergie potentielle du système peut s'écrire sous la forme :

$$U_{pot} = \int_{\mathcal{C}_a} \frac{1}{2} \omega_1(u) \left\| \frac{\partial \mathcal{C}_a}{\partial u} \right\|^2 du + \int_{\mathcal{C}_a} \frac{1}{2} \omega_2(u) \left\| \frac{\partial^2 \mathcal{C}_a}{\partial^2 u} \right\|^2 du \quad (8.5)$$

où les termes ω_1 et ω_2 permettent de contrôler l'extension du contour et sa courbure; généralement ces termes ne dépendent pas de u .

Dans le cas où le contour peut être décrit par une fonction linéaire des variables généralisées [Terzopoulos 91, Bascle 92], les équations d'évolution du système peuvent être décrites sous la forme matricielle suivante :

$$\Phi \ddot{\mathbf{a}} + \Upsilon \dot{\mathbf{a}} + \Psi \mathbf{a} = \Theta(\mathbf{a}) \quad (8.6)$$

où les matrices Φ , Υ , Ψ et Θ dépendent respectivement de l'énergie cinétique pour Φ , des forces de frottement fluides pour Υ , de l'énergie potentielle pour Ψ et enfin des forces extérieures généralisées calculées tout au long de \mathcal{C}_a pour Θ . Habituellement, les termes d'énergie sont négligés ($\mu = 0$).

\mathcal{C}_a est décrite sous une forme polaire :

$$\mathcal{C}_a(u) = \mathbf{x}_c + \rho_a(u) \begin{pmatrix} \cos(u) \\ \sin(u) \end{pmatrix} \quad (8.7)$$

où \mathbf{x}_c est le centre du contour et $\rho_a(u)$, le rayon polaire, est approximé par une décomposition en série de Fourier :

$$\rho_a(u) = a_0 + \sum_{i=1}^h a_i \cos(iu) + b_i \sin(iu) \quad (8.8)$$

où h est le nombre d'harmoniques.

Dans ce cas, les matrices Υ et Ψ impliquées dans (8.6) sont diagonales et constantes. Seul le terme Θ reste à calculer en ligne. Cette approche est donc particulièrement adaptée au temps réel. La rapidité de traitement de notre approche permet ainsi de suivre le contour dans une séquence d'images.

La seule force généralisée considérée est la force générée par l'image et appliquée au contour :

$$E_{cont} = - \int_{\mathcal{C}_a} \|\nabla I(\mathbf{x}(u))\| du \quad (8.9)$$

où $I(\mathbf{x}(u))$ est l'intensité du point $\mathbf{x}(u)$.

En pratique, la convergence des algorithmes de contour actif n'est obtenue que pour une initialisation du contour au voisinage du contour réel. Ce problème reste un problème ouvert et a pu être traité en introduisant, par exemple, des forces de pression [Cohen 93] ou par le biais d'un processus de diffusion [Xu 98].

L'approche proposée par C. Collewet considère un terme d'énergie potentielle dépendant de la surface intérieure délimitée par \mathcal{C}_a . Cette surface s'écrit très simplement grâce à la formule de Green-Riemann :

$$S_a = \pi(a_0^2 + \frac{1}{2} \sum_{i=1}^h a_i^2) \quad (8.10)$$

La contribution de ce terme est également une matrice diagonale et constante venant s'ajouter à la matrice Ψ . Cette façon de faire est, dans l'idée, similaire aux forces de pression introduites par Cohen mais les calculs sont ici beaucoup plus simples.

8.2.3 Résultats expérimentaux

Ce paragraphe présente les résultats de détection de contours et de suivis obtenus pour quelques objets du quotidien. La figure 8.7 présente l'évolution du contour actif depuis sa position initiale jusqu'à son état d'équilibre sur le contour de l'objet. Les points de ces contours sont ensuite utilisés pour estimer les coefficients des coniques dans les vues.

L'arrière plan uniforme autorise une initialisation grossière des contours actifs qui convergent vers les contours réels de l'objet même dans le cas d'une initialisation lointaine.

Cependant si l'objet est placé sur un fond très texturé, le contour actif doit être initialisé assez près du contour réel de l'objet pour ne pas diverger. L'initialisation des contours actifs est la pierre d'achoppement de cette méthode de détection de contour. Dans la section suivante des pistes de réflexion seront données pour aller vers une initialisation automatique.

Si l'emplacement de l'objet est suffisamment proche dans deux images successives d'une séquence vidéo, les contours détectés dans une des images peuvent servir d'initialisation pour la détection du contour de l'objet dans l'image suivante. Ainsi, l'objet peut être suivi dans une séquence de vues (voir la figure 8.2.3). Si l'écart entre deux vues est trop important, une initialisation par le contour précédent peut conduire à une mauvaise détection. La prochaine section propose des solutions simples pour résoudre ce problème lorsque l'objet est de couleur uniforme ou lorsque le fond est de couleur relativement uniforme.

8.3 Initialisation des contours actifs en un *clac*

L'objet de cette thèse est de proposer une aide technique à la saisie demandant à l'utilisateur un seul *clac* sur une vue déportée. Nous avons montré dans la première partie de cette thèse que l'information donnée par le *clac* suffit à la localisation de l'objet. Étant donné la position 3D de l'objet, cette position peut être utilisée pour projeter un point dans la vue de la caméra embarquée. Un point appartenant à l'objet est alors disponible et peut être utilisé pour initialiser le contour actif.

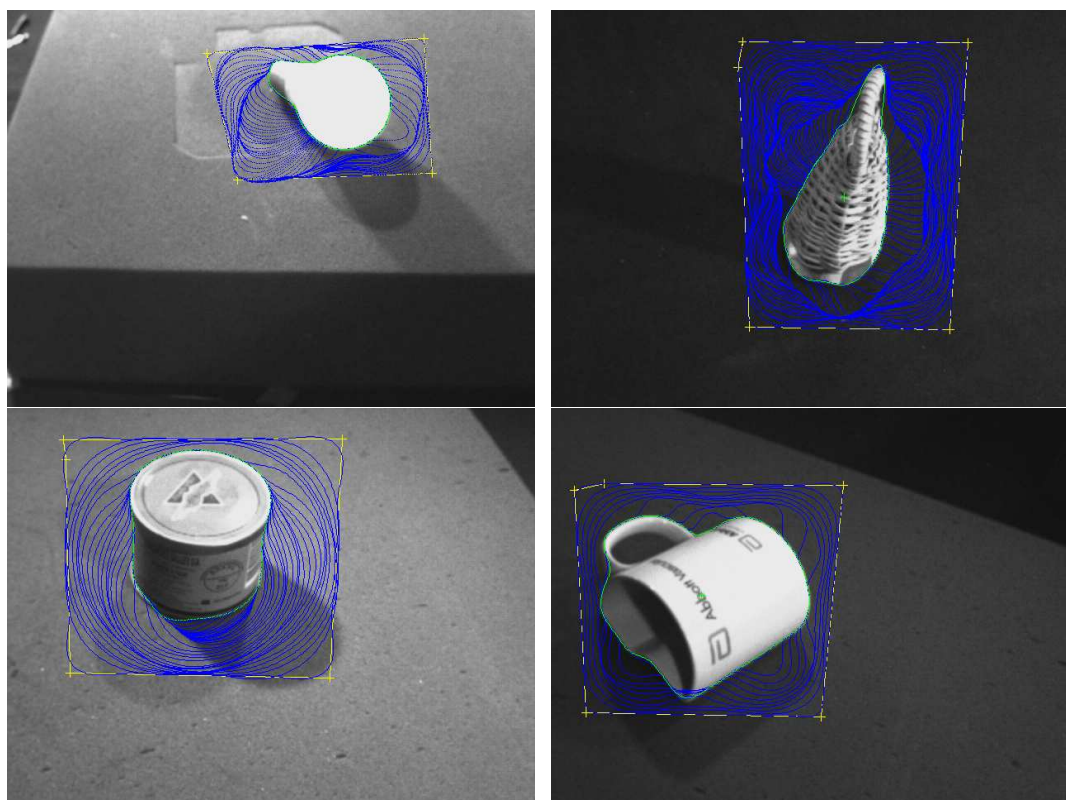


FIG. 8.7 – Détection de contours par contour actif. Les contours sont initialisés par la sélection de quatre points autour de l'objet. Le processus de détection converge au bout d'une cinquantaine d'itérations. Le contour initial est tracé en jaune. L'évolution du contour correspond à l'ensemble de courbes bleues, le contour à convergence est une courbe verte.

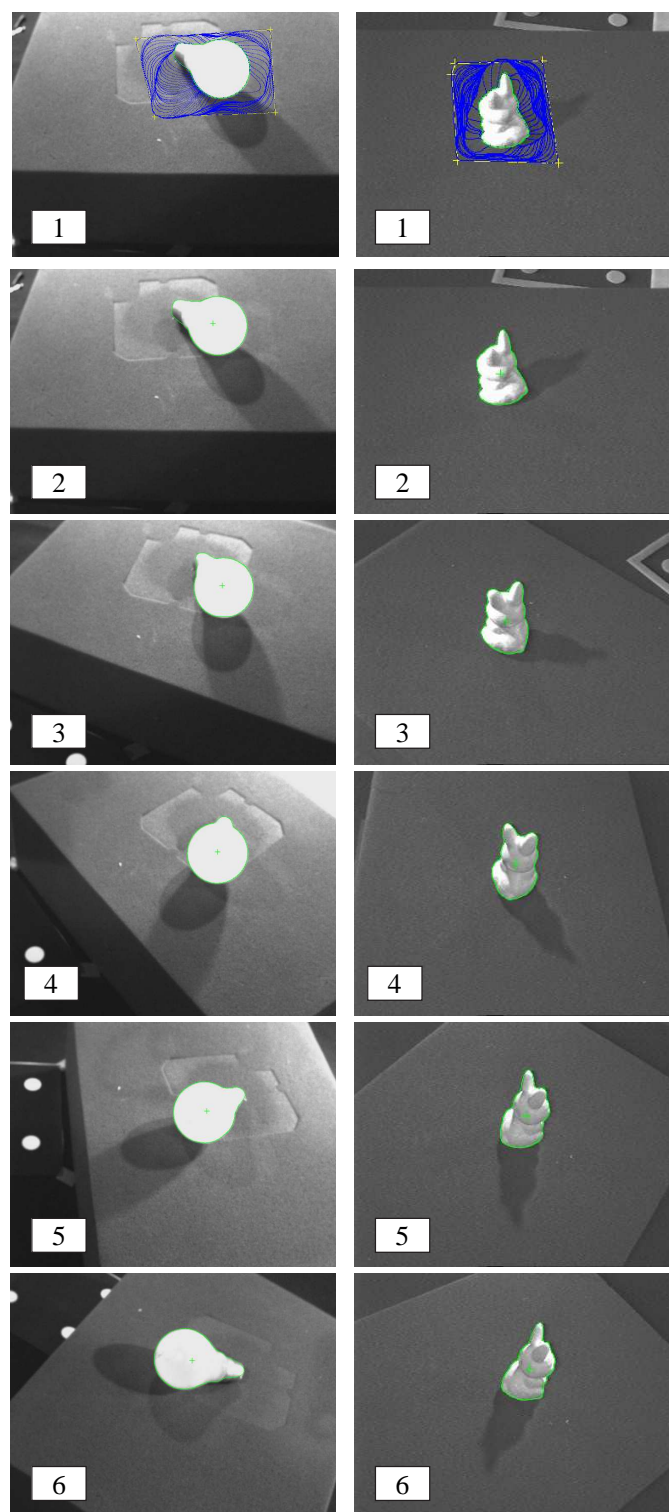


FIG. 8.8 – Suivi du contour dans une séquence d'images. Dans la première image, le contour est initialisé par quatre points et converge vers le contour réel de l'objet. Dans les images suivantes de la séquence, les contours sont initialisés en utilisant le contour détecté dans l'image précédente. La méthode de suivi est robuste à des déplacements relativement faible de l'objet entre deux vues.

L'initialisation des contours actifs est une étape cruciale à la convergence de l'algorithme de détection de l'objet, en particulier, si l'objet est placé sur un fond texturé. Dans ce paragraphe, nous proposons quelques axes de réflexion pour aller vers une initialisation des contours actifs à partir des coordonnées d'un seul point de l'objet.

8.3.1 Initialisation à partir d'un point pour la détection d'un objet uniforme

Si l'objet est de couleur relativement uniforme, les contours actifs peuvent être initialisés à l'intérieur de l'objet. Le point est utilisé pour définir un petit rectangle (voir l'image de gauche de la figure 8.9).

Une fois le contour de l'objet détecté dans la première image d'une séquence, les coordonnées de son barycentre peuvent être utilisées pour mettre à jour les coordonnées du *clac*. L'objet est ainsi suivi dans la séquence 8.9. Dans cette séquence, le terme d'extension est nul, le terme de courbure vaut 10^{-3} et le terme de surface est négatif et vaut $-0,6$. Lorsque le contour est initialisé à l'extérieur de l'objet, ce terme est positif.

Dans les images 1, 2 et 3 de la séquence, un petit décrochage peut être remarqué à l'endroit de la concavité entre l'oreille et le dos du chat. Il est dû à la définition polaire du contour actif : si plusieurs points de contours se trouvent sur un même rayon issu du centre du contour actif, alors le point de contour le plus éloigné du centre est détecté. Ce problème pourrait être résolu en représentant les contours par une fonction paramétrique basée sur les descripteurs de Fourier qui permettrait de prendre en compte les concavités. Dans le cadre de notre étude, ce type d'imprécision n'est cependant pas crucial puisque le contour est utilisé pour calculer les coefficients d'une conique englobante qui ne permet pas de représenter les concavités.

8.3.2 Initialisation et suivi en utilisant une boîte englobante

Si l'objet est isolé sur un fond relativement uniforme et en supposant qu'il est entièrement inclus dans l'image (voir l'annexe C), une solution est de choisir pour contour initial un rectangle centré sur le *clac* ayant pour côté la distance séparant ce point du contour le plus proche. Le terme de surface est positif.

La figure 8.10 présente deux résultats de détection des contours de l'objet obtenus en initialisant les contours actifs par une boîte englobante.

Cette méthode peut être utilisée pour suivre un objet placé sur un fond uniforme dans une séquence d'images au cours de laquelle les déplacements de l'objet dans deux vues successives sont importants. Un exemple d'un tel suivi est donné par la figure 8.11.

Le nombre d'itérations pour obtenir la convergence est alors d'autant plus important que l'échelle de l'objet est petite relativement aux dimensions de l'image. Cependant, si l'échelle de l'objet est donnée, alors il est possible de diminuer le nombre d'itérations nécessaires en initialisant le contour actif sur un rectangle dont le côté est approximativement l'échelle de l'objet.

8.3.3 Évaluation de l'échelle caractéristique de l'objet

Plutôt que d'utiliser une boîte englobante dimensionnée en fonction de la distance aux bords de l'image, une meilleure solution est de dimensionner cette boîte pour qu'elle corresponde aux dimensions et à l'orientation de l'objet. Les dimensions pixeliques de l'objet peuvent être évaluées par le biais du calcul de l'échelle caractéristique d'un point. Ici, le point correspond à la projection de la localisation 3D estimée de l'objet. Le calcul de l'échelle caractéristique en un point et ses propriétés ont été intensivement étudiées par Lindeberg [Lindeberg 98].

L'estimation de l'échelle d'une structure locale est calculée au voisinage d'un point en sélectionnant l'échelle σ_n qui correspond au maximum d'une certaine fonction, définie sur l'espace d'échelles, l'opérateur d'échelle. L'échelle pour laquelle cet opérateur est maximal est l'échelle caractéristique

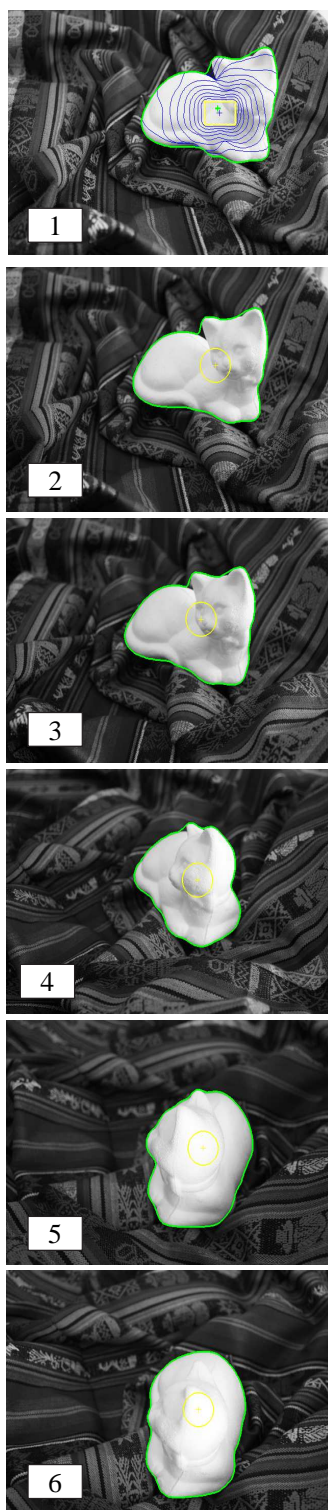


FIG. 8.9 – Suivi d'un objet de couleur uniforme sur une arrière plan complexe. Le contour actif est initialisé à l'intérieur de l'objet, par un petit rectangle centré sur le clic. Le centre du contour est mis à jour après convergence et est utilisé comme point d'initialisation pour la détection du contour dans l'image suivante de la séquence. Dans l'image 1) le contour jaune est le contour initial, les contours bleus sont les positions intermédiaires et le contour vert est le contour à convergence.

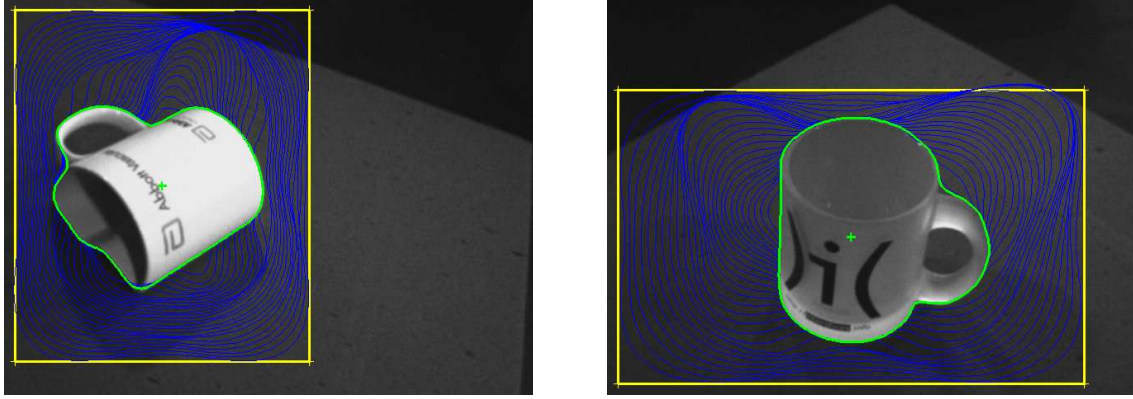


FIG. 8.10 – Deux exemples d'initialisation à partir d'un rectangle centré sur le clic et de côté la distance au bord le plus proche. Les objets sont placés sur un fond uniforme. Le rectangle jaune est le contour initial, les courbes bleues sont les contours intermédiaires et le contour vert est le contour à la convergence.

ou l'échelle intrinsèque de la structure. Le terme *caractéristique* a été choisi car l'échelle *caractéristique* correspond à la longueur des structures au voisinage du point. L'opérateur d'échelle peut avoir plusieurs maxima locaux qui correspondent à des structures concentriques centrées sur le point.

Ainsi, si la structure la plus forte au voisinage du point est le contour de l'objet, le calcul de l'échelle caractéristique permettra d'estimer la taille de l'objet dans l'image.

Parmi les opérateurs les plus courants, on trouve :

- le maximum du Laplacien of gaussien LoG [Lindeberg 98]
- le maximum du déterminant de la matrice des moments 2D [Gardin 94]
- le maximum de la trace de la matrice des moments 2D [Gardin 94]

Maximum du LoG

$$|\text{LoG}(i, j, \sigma_n)| = \sigma_n^2 |L_{xx}(i, j, \sigma_n) + L_{yy}(i, j, \sigma_n)| \quad (8.11)$$

Cet opérateur mesure l'échelle à laquelle la similarité est maximale entre la structure locale de l'image et la structure de l'opérateur Laplacien. Cette mesure est donc très bien adaptée à la détection de blobs : lorsque la taille du noyau de convolution du LoG correspond à la taille d'un blob centré sur le point désigné, la réponse est maximale. Les figures 8.12 et 8.13 présentent la détection de l'échelle caractéristique en utilisant le maximum du LoG.

Matrice des moments 2D La matrice des moments du second degré d'un point peut être vue comme la matrice de covariance d'une variable aléatoire de dimension 2 ou, pour utiliser une analogie mécanique, elle représente la matrice des moments d'inertie d'une distribution massique sur le plan. Cette matrice peut être représentée graphiquement comme une ellipse. Elle est utilisée dans le calcul des points de Harris [Harris 88].

Soit I l'intensité lumineuse des pixels de l'image et $\nabla = (I_x, I_y)^\top$ le gradient de l'image. La matrice des moments du second degré est définie comme suit :

$$\mu_I(\mathbf{x}, \sigma_i, \sigma_d) = \begin{pmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{pmatrix} \quad (8.12)$$

$$= g(\sigma_i) * \begin{pmatrix} I_x^2(\mathbf{x}, \sigma_d) & I_y I_x(\mathbf{x}, \sigma_d) \\ I_y I_x(\mathbf{x}, \sigma_d) & I_y^2(\mathbf{x}, \sigma_d) \end{pmatrix} \quad (8.13)$$

où σ_i est l'échelle d'intégration et σ_d est l'échelle de différentiation.

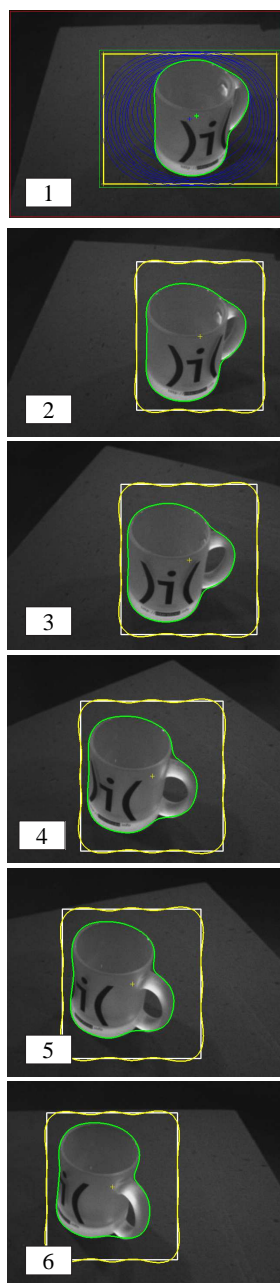


FIG. 8.11 – Suivi d'un objet sur un fond uniforme. Le contour actif est initialisé à l'extérieur de l'objet, par une boîte englobante centrée sur un point appartenant à l'objet et de côté, deux fois la distance séparant ce point des bords de l'image. Après convergence, le point central du contour est mis à jour en calculant le barycentre du contour et la boîte englobante correspondant. Cette nouvelle boîte est utilisée comme initialisation pour l'image suivante.

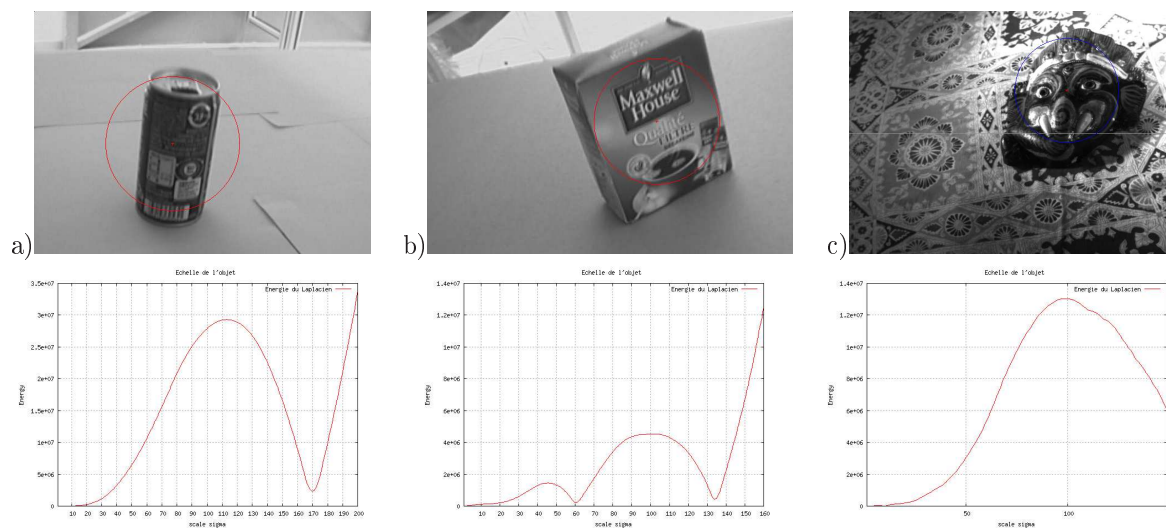


FIG. 8.12 – Détection de l'échelle d'un seul objet en évaluant le maximum du LoG.

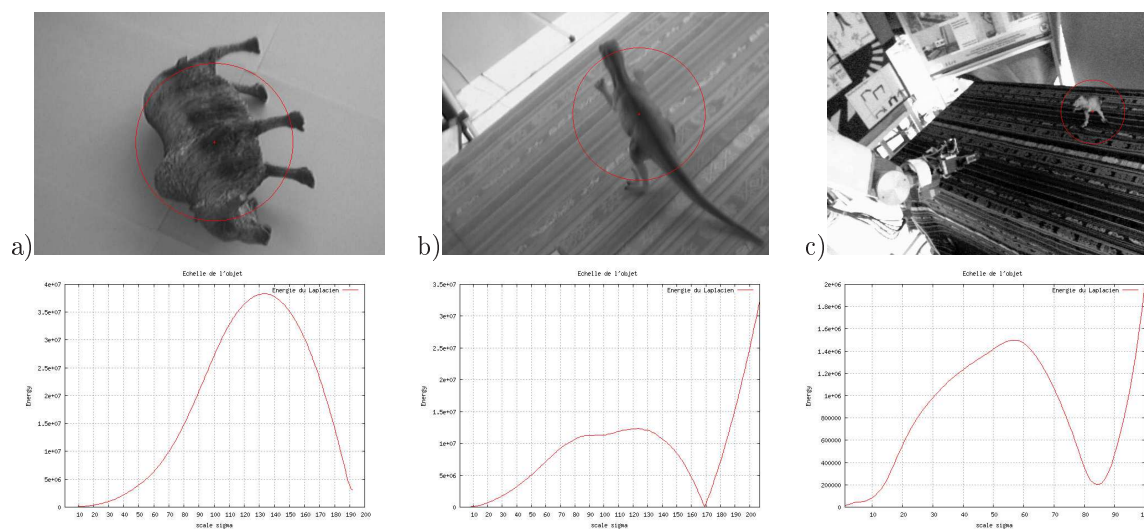


FIG. 8.13 – Détection de l'échelle d'un seul objet en évaluant le maximum du LoG.



FIG. 8.14 – Détection de l'échelle d'objets composant une scène.

Cette matrice est invariante aux translations et il peut être montré que sa trace et son déterminant sont invariants aux rotations et aux translations [Gardin 94]. D'autre part, un changement d'échelle spatialement uniforme et les transformations affines de l'intensité lumineuse affectent μ uniquement d'un facteur d'échelle. La fonction $g(\sigma_i)$ est un opérateur de moyenne, typiquement, un filtre gaussien. La figure 8.15 présente les résultats de sélection d'échelle obtenus en utilisant la trace et le déterminant de μ_I . Le maximum de ces opérateur est trouvé pour une même échelle, légèrement inférieure à celle détectée en utilisant le LoG.

Il a été montré que la détermination de l'échelle caractéristique par maximisation du LoG permet d'obtenir un meilleur pourcentage de bonnes estimations que la trace ou le déterminant de μ_I [Mikolajczyk 06]. D'autre part, le calcul de μ_I est plus coûteux que le calcul du LoG à différentes échelles. Ainsi, nous estimerons l'échelle de l'objet en cherchant le maximum du LoG. Cependant, la matrice des moments 2D est particulièrement intéressante pour initialiser le contour de l'objet car elle permet d'orienter la structure et de la représenter par une ellipse.

8.3.4 Orientation de l'objet

L'évaluation de l'échelle caractéristique en un point permet de calculer le rayon d'un cercle centré sur le point et qui correspond au mieux au contour de lus fort contraste. Cette échelle peut donc être utilisée pour dessiner une boîte carrée centrée sur le point qui servira à initialiser les contours actif. Ce type d'initialisation fonctionne très bien lorsque l'objet considéré est de forme approximativement circulaire. Cependant, dans le cas d'un objet de forme allongé, le contour ne peut pas être initialisé de cette manière.

Le calcul de la matrice des moments du second degré à l'échelle caractéristique apporte une solution au problème de l'initialisation du contour de ces objets. Elle permet d'estimer l'orientation et les dimensions 2D de l'objet, en d'autres termes, le calcul de cette matrice permet d'estimer les paramètre

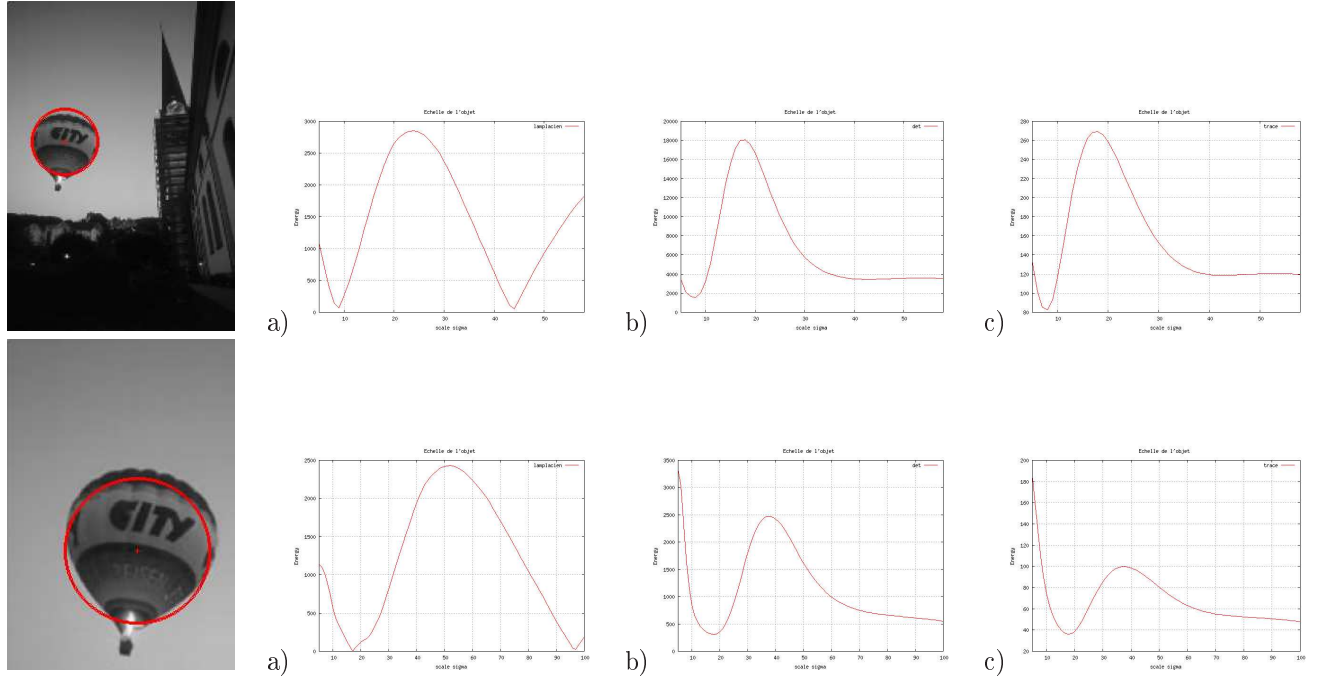


FIG. 8.15 – Évaluation de trois détecteurs d'échelle, la norme du laplacien (a) et le déterminant (b) et la trace (c) de la matrice des moments 2D pour différentes échelles.

de l'ellipse qui correspond au mieux aux contours de l'objet. Nous verrons dans ce paragraphe que l'ellipse ainsi définie peut servir à l'initialisation du contour.

La matrice des moments du second degré est symétrique et semi définie positive, alors, l'équation suivante définit une ellipse de centre \mathbf{x}_c (voir figure 8.16) :

$$(\mathbf{x} - \mathbf{x}_c)^T \boldsymbol{\mu}_I (\mathbf{x} - \mathbf{x}_c) = 1 \quad (8.14)$$

Le calcul des valeurs propres (λ_{min} , λ_{max}) et des vecteurs propres de $\boldsymbol{\mu}_I$ permet de calculer l'orientation de l'ellipse et ses dimensions à un facteur près.

Comme proposé dans [Mikolajczyk 04], l'échelle caractéristique de l'objet peut être calculée en utilisant le maximum du LoG et utilisée ensuite pour calculer la matrice des moments 2D $\boldsymbol{\mu}_I$ correspondant à cette échelle. La variance d'intégration σ_i est alors fixée à $1.5\sigma_n$ et σ_d est compris dans l'intervalle $[0.5\sigma_i, 0.75\sigma_i]$ et choisie de sorte à maximiser l'isotropie de l'ellipse, c'est-à-dire à maximiser le rapport $\lambda_{min}/\lambda_{max}$. Lorsque ce rapport vaut 1, l'ellipse est un cercle. Lorsqu'il vaut zéro ou l'infini, c'est une droite. Finalement, pour calculer les dimensions de l'ellipse, le petit axe est fixé à la valeur $2\sigma_n$ et le plus grand axe est fixé à $2\lambda_{max}/\lambda_{min}\sigma_n$.

La figure 8.17 présente quelques résultats obtenus en détectant l'échelle par la recherche du maximum du Laplacien puis en détectant l'ellipse associée à cette échelle grâce au calcul de la matrice des moments du second degré.

Les contours actifs peuvent être initialisés à partir d'une ellipse ou d'une boîte englobante de dimensions $s\frac{1}{\lambda_{min}}$ et $s\frac{1}{\lambda_{max}}$, avec $s > 2$ pour un contour initial à l'extérieur de l'objet ou $s < 2$ pour un contour initial à l'intérieur de l'objet.

Notons cependant que le calcul de l'échelle intrinsèque est très sensible à la position du point sélectionné (voir la figure 8.18,a)). D'autre part, si le point n'est pas centré sur l'objet, alors l'ellipse obtenue est décalée par rapport à l'objet et s'avère une très mauvaise initialisation pour le contour actif (voir la figure 8.18,b)). Ainsi, plutôt que d'utiliser directement la projection de la position estimée comme centre de l'estimation de l'échelle caractéristique de l'objet, il est préférable de recentrer le

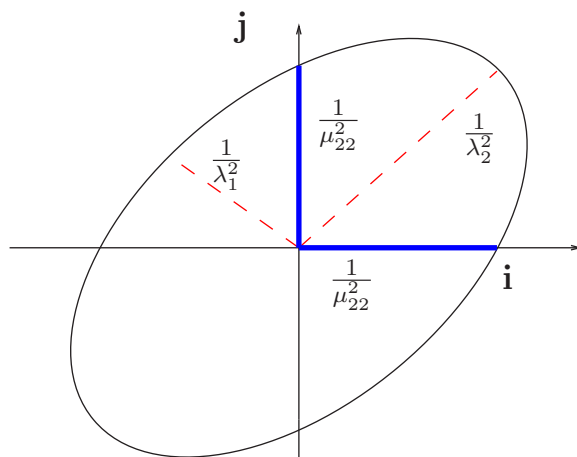


FIG. 8.16 – Estimation de l'échelle et de l'orientation d'une boîte

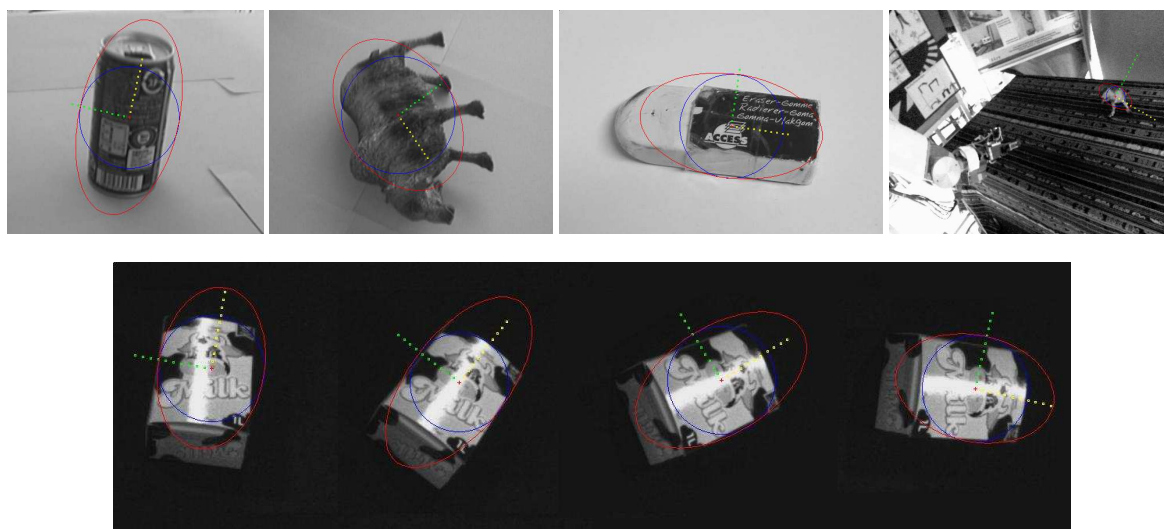


FIG. 8.17 – Estimation de l'échelle et de l'orientation pour différents objets. Pour tous les objets testés, un cercle bleu est centré sur le point sélectionné. Son rayon est l'échelle caractéristique en ce point. L'ellipse rouge et ses deux axes vert et jaune sont calculés à partir de la matrice des moment 2D évaluée au point sélectionné et à l'échelle caractéristique.

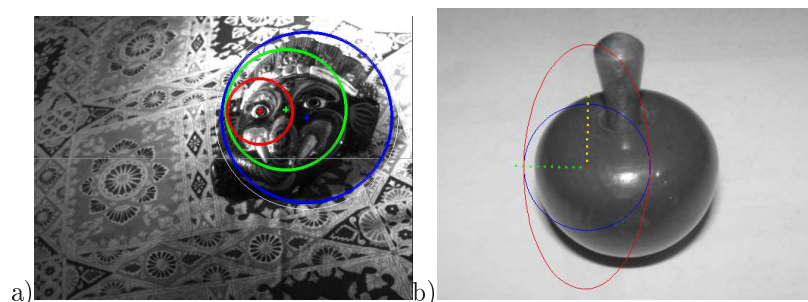


FIG. 8.18 – La valeur de l'échelle caractéristique dépend de la position du point pour laquelle elle est calculée.

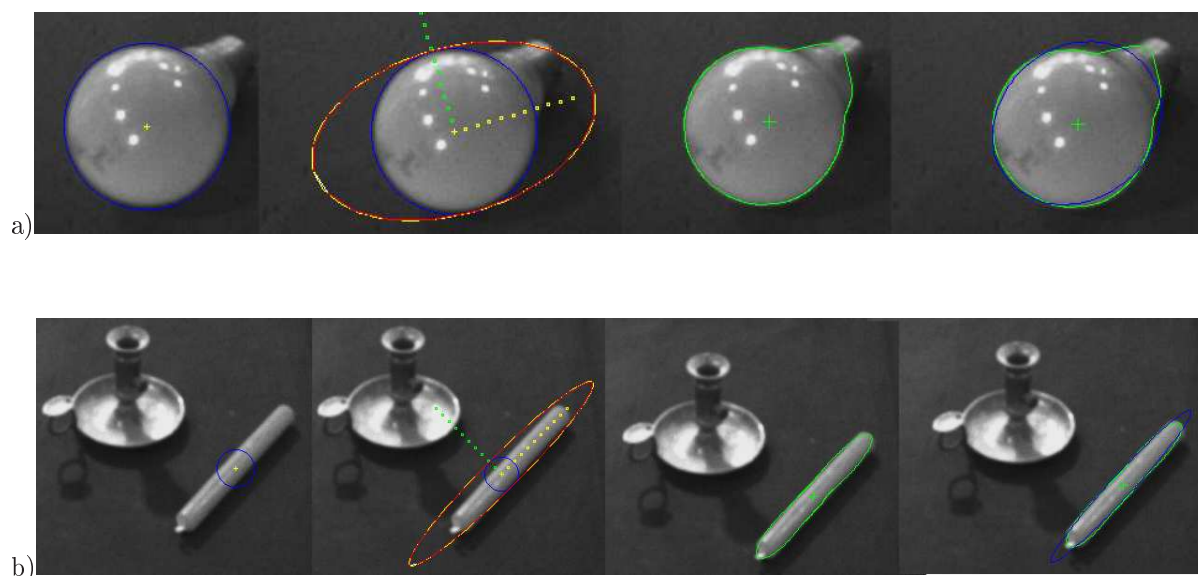


FIG. 8.19 – deux exemples.

point dans un premier temps en recherchant le maximum du LoG sur l'espace d'échelle le plus proche du point initial [Mikolajczyk 04].

La figure 8.19 présente deux exemples dans lesquels les contours actifs sont initialisés à partir d'un seul point. Dans la première série d'images, l'objet est de forme approximativement sphérique et le contour actif aurait pu être initialisé directement à partir de la ellipse définie par l'échelle caractéristique. À l'opposé, dans le cas du second objet, qui est un cylindre très allongé, une initialisation par le petit cercle bleu est très mauvaise, alors qu'une initialisation par l'ellipse dimensionnée par le calcul des moments 2D permet d'obtenir le contour réel de l'objet.

8.4 Conclusion

La méthode proposée permet de détecter un objet uniforme sur un arrière plan quelconque ou un objet quelconque sur un fond uniforme à partir d'une initialisation grossière. Dans ces deux cas, la méthode peut être initialisée en un seul *clic*.

Si l'objet est texturé sur un fond texturé, le contour initial doit être proche du contour réel pour garantir la convergence du contour actif. Le calcul de l'échelle caractéristique de l'objet et l'estimation

de son orientation dans l'image est une piste à explorer pour résoudre ce problème. Elle pourrait être une solution peu coûteuse à l'initialisation en un seul *clic* des contours actifs.

La méthode proposée permet de suivre l'objet dans une séquence d'images. Si l'objet est bien détecté dans la première vue et si l'écart entre les contours de l'objet dans deux vues successives est faible alors, la détection de contour est très robuste aux défauts d'illumination. En revanche, si l'écart entre deux vues successives est important, le contour actif ne peut pas être initialisé à partir du résultat de la vue précédente. À nouveau, la méthode de détection de l'échelle et de l'orientation pourrait être une solution pour réinitialiser le contour actif s'il diverge.

La partie suivante présente une méthode par vision active pour sélectionner au mieux les vues afin d'obtenir la meilleure estimation des paramètres de la quadrique correspondant à la forme de l'objet.

Sélectionner les meilleures vues : une méthode de vision active

Lors de l'estimation de la forme d'un objet par une fonction quadratique, au terme du processus d'optimisation, les résidus sont rarement tous nuls. L'erreur sur l'identification de paramètres peut provenir de plusieurs sources :

- **Le bruit de mesure** : traitement de l'image, erreurs d'étalonnage du système de vision, par exemple.
- **L'erreur de modélisation** : la forme de l'objet ne correspond pas à une quadrique
- **L'erreur d'estimation** : les paramètres n'ont pas atteint l'optimum.

Il est difficile d'agir sur les deux premières causes d'erreur. Notre approche permet d'approximer la forme d'un objet par une fonction quadratique, si la forme de l'objet ne correspond pas suffisamment à ce modèle, l'estimation peut échouer (voir les hypothèses 5.1). D'autre part, nous tentons de rendre cette méthode robuste aux bruits de mesure, nous ne pouvons cependant pas faire chuter les résidus en deçà d'un certain seuil en utilisant des méthodes des moindres carrés. Par ailleurs, en négligeant le bruit et en considérant un objet qui correspond parfaitement au modèle, une autre cause d'erreur subsiste. Il s'agit de l'ambiguïté entre tous les espaces disjoints de l'espace des paramètres qui résolvent l'équation $\chi^2(\mathbf{q}) = 0$. Cette erreur d'estimation des paramètres peut être nettement diminuée par un choix judicieux des vues. Ainsi, la qualité de reconstruction peut être améliorée.

Intuitivement, il apparaît que si N vues proches et d'orientations similaires donnent une bonne estimation de la "largeur" et de la "hauteur" de l'objet, elles donnent une estimation pauvre de sa "profondeur". L'idée d'acquérir une ou plusieurs images supplémentaires pour rassembler des informations pertinentes à l'amélioration de l'estimation des paramètres est naturelle. Les meilleures vues sont celles qui apporteraient les informations les plus nouvelles en comparaison des informations disponibles grâce au traitement des vues déjà acquises. Cette idée s'inspire des travaux de Whaite et Ferrie qui visent à doter un système robotique de "curiosité" [Whaite 97b, Whaite 97a], en quelque sorte, le pousser à observer les parties de l'objet qui sont les moins bien modélisées. Cette approche repose essentiellement sur la théorie de l'information en traitement du signal et sur le résultat de l'optimisation des moindres carrés. La méthode que nous proposons s'approche de la méthode proposée par [Flandin 02] qui consiste à déplacer une caméra embarquée de façon à minimiser l'incertitude sur les paramètres d'un ellipsoïde englobant un objet.

Dans cette section, nous proposons un critère objectif permettant de sélectionner les meilleures vues pour obtenir une reconstruction sans ambiguïté de la quadrique englobant l'objet. Le premier paragraphe développe le calcul de l'incertitude sur les paramètres en fin de processus d'estimation. Le second paragraphe montre comment évaluer l'incertitude associée aux différentes vues, c'est-à-dire déterminer celles qui sont susceptibles d'apporter la plus grande quantité d'informations. Cette mesure peut être directement utilisée pour contrôler le robot vers les prochaines meilleures vues.

9.1 Information contenue dans une vue

Les méthodes de vision active consiste à déplacer un capteur en vue d'améliorer un critère lié à une tâche haut niveau. Cette tâche peut être une tâche de reconstruction. Le critère à minimiser est alors souvent l'incertitude sur un ensemble de paramètres. Il peut également s'agir d'une tâche d'exploration ou encore d'une tout autre tâche robotique.

Dans le cadre de cette étude, la tâche à mener à bien est la caractérisation d'un objet en vue de sa saisie, c'est-à-dire estimer sa position, son orientation et ses dimensions.

9.1.1 Incertitude sur les paramètres

Les paramètres de la quadrique correspondant à la surface de l'objet sont estimés en utilisant un processus des moindres carrés itératif. À partir des résultats de cette optimisation numérique, il est possible d'évaluer l'incertitude sur l'estimation des paramètres de la quadrique.

Au voisinage de l'optimum, la fonction de coût est approximée par une forme quadratique. Une fois qu'un minimum acceptable de la fonction de coût $\chi^2(\mathbf{q})$ a été trouvé pour un jeu de paramètres \mathbf{q}_{min} , la variation de χ^2 autour de ce minimum pour une variation $\delta\mathbf{q}$ des paramètres ajustés \mathbf{q}_{min} est donnée par :

$$\chi^2(\mathbf{q}_{min} + \delta\mathbf{q}) = \chi^2(\mathbf{q}_{min}) + \nabla\mathbf{F} \delta\mathbf{q} + \delta\mathbf{q}^\top \mathbf{G} \delta\mathbf{q} + o(|\delta\mathbf{q}|^2) \quad (9.1)$$

où $\delta\mathbf{q} = \mathbf{q}_{min} - \mathbf{q}$ est un petit déplacement depuis le minimum. Par définition, au minimum $\nabla\mathbf{F} = 0$ et $\delta\mathbf{q}^\top \mathbf{G} \delta\mathbf{q} > 0$. Pour des très petits déplacements $\delta\mathbf{q}$, l'équation (9.1) peut se réécrire :

$$\begin{aligned} \Delta\chi^2(\delta\mathbf{q}) &= \chi^2(\mathbf{q}_{min} + \delta\mathbf{q}) - \chi^2(\mathbf{q}_{min}) \\ &\sim \delta\mathbf{q}^\top \mathbf{G} \delta\mathbf{q} \end{aligned}$$

Aucune composante de $\Delta\chi^2(\delta\mathbf{q})$ n'est nulle *a priori* et il peut être montré que $\Delta\chi^2(\delta\mathbf{q})$ est distribué comme une loi chi-carré d'une variable aléatoire à distribution normale. La matrice de covariance définit un ellipsoïde de confiance sur l'espace des paramètres. Cet ellipsoïde entoure les paramètres réels $\gamma\%$ du temps. Autrement dit :

- $\Delta\chi^2(\delta\mathbf{q}) < 1$ pour $\delta q < 1\sigma$, dans $\gamma = 68.3\%$ des cas ;
- $\Delta\chi^2(\delta\mathbf{q}) < 4$ pour $\delta q < 2\sigma$, dans $\gamma = 95.4\%$ des cas ;
- $\Delta\chi^2(\delta\mathbf{q}) < 9$ pour $\delta q < 3\sigma$, dans $\gamma = 99.73\%$ des cas ;
- etc.

Intéressons nous plus particulièrement à la variation de χ^2 autour du minimum $\chi^2(\mathbf{q}_{min})$ lorsqu'un seul paramètre q_i varie arbitrairement, les autres paramètres restant fixés à leur valeur estimée de \mathbf{q}_{min} .

Soit M le nombre de paramètres, notons χ_{M-1}^2 le chi-carré à $M-1$ degrés de liberté obtenu en fixant le paramètre q_i à une valeur arbitraire. Soit \mathbf{q}_{M-1} le nouveau jeu de paramètres qui minimise χ_{M-1}^2 . Posons $\Delta\chi_i^2 = \chi_{M-1}^2(\mathbf{q}_{M-1}) - \chi^2(\mathbf{q}_{min})$ et $\delta\mathbf{q} = \mathbf{q}_{M-1} - \mathbf{q}_{min}$. L'incertitude δq_i sur le paramètre q_i peut être reliée à $\Delta\chi^2(q_i)$ en remarquant que $\nabla\mathbf{F}(\mathbf{q}_{M-1}) = 0$ sur toutes les composantes sauf la i ème :

$$\delta q_i = -\frac{\partial\chi^2}{\partial q_i} C_{ii}$$

En introduisant ce résultat dans l'équation (9.1), il en est déduit que :

$$\Delta\chi_i^2 = \delta\mathbf{q}^\top \mathbf{G} \delta\mathbf{q} = \frac{(\delta q_i)^2}{C_{ii}}$$

et

$$\delta q_i = \pm \sqrt{\Delta\chi_i^2} \sqrt{C_{ii}}$$

Ainsi, si l'incertitude sur le paramètre q_i est définie par 1σ , soit $\Delta\chi_i^2 = 1$, on obtient la même relation que dans un cas linéaire :

$$\delta q_i = \sqrt{C_{ii}} \quad (9.2)$$

9.1.2 Critère de sélection de la vue suivante

Intuitivement, on pressent qu'un choix judicieux des vues peut améliorer les paramètres qui présentent une grande incertitude. La question est de trouver un critère objectif qui permette de quantifier l'information contenue dans une vue. Ce critère peut être basé sur la théorie de l'information développée par Shannon, comme le propose [Whaite 97b]. L'information contenue dans une vue dépend de la valeur du déterminant de la matrice de covariance et [Whaite 97b] montre que maximiser l'information contenue dans une vue revient à minimiser le déterminant de la matrice de covariance.

Ainsi, la clé pour communiquer l'incertitude associée à une vue au système haut niveau qui gère les déplacements de la caméra est la nature de la surface chi-carré de l'incertitude dans l'espace des paramètres [Whaite 91]. L'incertitude peut être communiquée *via* la matrice de covariance $\mathbf{C} = \sigma^2 \mathbf{G}^{-1}$. Elle est immédiatement trouvée en évaluant la matrice de Gauss \mathbf{G}^{-1} à convergence du processus d'optimisation numérique qui permet d'estimer les paramètres de la quadrique (voir la section 7.3). Dans le cadre d'un processus itératif de calcul des paramètres, il est intéressant d'établir la relation entre la valeur de la covariance de la vue suivante et la valeur courante de la covariance. Ainsi, elle pourra être calculée incrémentalement et efficacement à chaque pas.

Cette matrice ne dépend que des positions occupées par la caméra au cours de la reconstruction et des paramètres estimés \mathbf{q}_{min} . Elle peut être interprétée, géométriquement, comme le volume de l'ellipsoïde de confiance dans l'espace des paramètres qui vaut $\det \mathbf{C}$. Pour réduire l'erreur sur les paramètres, il faut trouver la vue qui minimise le volume de l'incertitude $\det \mathbf{C}$. En notant \mathbf{C}_j la matrice de covariance de la j ème vue et sachant que $\mathbf{C}_j = \mathbf{G}_j^{-1}$, quelle doit être la prochaine pose de la caméra pour que $\det \mathbf{C}_{j+1}$ soit minimale? Répondre à cette question revient à chercher la vue qui donne le $\det \mathbf{G}_{j+1}$ maximal?

Soit $\mathbf{G}_j = \mathbf{J}_{C_j}^\top \mathbf{J}_{C_j}$ où le jacobien est construit par empilement des jacobiens associés à chaque vue $\mathbf{J}_{C_j} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_j)^\top$. Si on parvient à exprimer la matrice de Gauss *a priori* de la prochaine vue en fonction des paramètres courants et de la matrice de Gauss courante, il est alors possible de chercher la pose qui permet de maximiser le $\det \mathbf{G}_{j+1}$.

$$\mathbf{G}_{j+1} = \mathbf{G}_j + \mathbf{g}_{j+1}^\top \mathbf{g}_{j+1} \quad (9.3)$$

et

$$\det(\mathbf{G}_{j+1}) = \det(\mathbf{G}_j + \mathbf{g}_{j+1}^\top \mathbf{g}_{j+1}) \quad (9.4)$$

$$= \det(\mathbf{I} + \mathbf{g}_{j+1}^\top \mathbf{g}_{j+1} \mathbf{G}_j^{-1}) \det(\mathbf{G}_j) \quad (9.5)$$

Cette expression peut être transformée en considérant les propriétés des valeurs propres.

Définition : Si λ est une valeur propre de la matrice \mathbf{A} alors, il existe un vecteur \mathbf{x} non nul, tel que $\mathbf{Ax} = \lambda\mathbf{x}$.

À partir de cette définition, trois propriétés peuvent être déduites¹ :

1. si λ est une valeur propre de \mathbf{A} , alors $1 + \lambda$ est une valeur propre de $\mathbf{I} + \mathbf{A}$.
2. si \mathbf{B} est une matrice telle que $\dim(B) = n \times m$ et \mathbf{A} est une matrice de dimensions $m \times n$ alors les valeurs propres non nulles de \mathbf{BA} sont égales aux valeurs propres non nulles de \mathbf{AB}^\top
3. si \mathbf{B} est une matrice telle que $\dim(B) = n \times m$ et \mathbf{A} est une matrice carrée de dimensions $m \times m$ semi définie positive alors $\mathbf{B}^\top\mathbf{AB}$ est une matrice symétrique semi définie positive. Toutes ses valeurs propres sont donc positives ou nulles.

En prenant en compte ces trois propriétés, la fonction à maximiser (9.5) peut être réécrite :

$$\begin{aligned} \det G_{j+1} &= \det(\mathbf{I} + \mathbf{g}_{j+1}^\top \mathbf{g}_{j+1} \mathbf{G}_j^{-1}) \det(\mathbf{G}_j) \\ &= \det(\mathbf{I} + \mathbf{g}_{j+1} \mathbf{G}_j^{-1} \mathbf{g}_{j+1}^\top) \det(\mathbf{G}_j) \end{aligned}$$

La matrice $\mathbf{g}_{j+1} \mathbf{G}_j^{-1} \mathbf{g}_{j+1}^\top$ à une signification particulière, elle représente la covariance prédite pour la vue suivante en tenant compte de la variance courante. En notant λ_i , $i \in [1, 6]$ ses valeurs propres, l'expression de son déterminant peut être écrite comme suit :

$$\det(\mathbf{I} + \mathbf{g}_{j+1} \mathbf{G}_j^{-1} \mathbf{g}_{j+1}^\top) = \prod_{i=1}^6 (1 + \lambda_i)$$

d'où

$$\det \mathbf{G}_{j+1} = \det \mathbf{G}_j \prod_{i=1}^6 (1 + \lambda_i) \quad (9.6)$$

Comme \mathbf{G}_j est semi définie positive, $\mathbf{g}_{j+1} \mathbf{G}_j^{-1} \mathbf{g}_{j+1}^\top$ est également semi définie positive. Ses valeurs propres sont donc positives ou nulles. Ainsi, $\prod_{i=1}^6 (1 + \lambda_i) \geq 1$ et à chaque pas, on peut écrire :

$$\frac{\det(\mathbf{G}_{j+1})}{\det(\mathbf{G}_j)} = \prod_{i=1}^6 (1 + \lambda_i) > 1 \quad (9.7)$$

Cette égalité signifie qu'à chaque fois qu'une vue est ajoutée, quelle que soit sa position et les valeurs des paramètres courants, l'information augmente. Pour trouver la vue qui maximise l'information, il faut trouver la vue qui maximise $\det(\mathbf{I} + \mathbf{g}_{j+1} \mathbf{G}_j^{-1} \mathbf{g}_{j+1}^\top) = \prod_{i=1}^6 (1 + \lambda_i)$.

9.2 Calcul de la position suivante

L'approximation quadratique de la fonction de coût, sur laquelle repose le calcul de la covariance et le critère de sélection de la vue suivante, n'est vraie que localement autour de la position courante. Ainsi, la meilleure vue suivante est la vue qui permet de maximiser le critère $\det(\mathbf{I} + \mathbf{g}_{j+1} \mathbf{G}_j^{-1} \mathbf{g}_{j+1}^\top)$ au voisinage de la vue courante [Whaite 97b].

Une fois la meilleure vue calculée au voisinage de la vue courante, la caméra est déplacée dans la direction de cette vue tout en étant maintenue sur la sphère de vision. En d'autres termes, la caméra suit le gradient numérique de $\det(\mathbf{I} + \mathbf{g}_{j+1} \mathbf{G}_j^{-1} \mathbf{g}_{j+1}^\top)$. Une fois la caméra positionnée, le processus

¹Les démonstrations de ces propriétés sont données dans l'annexe D

d'estimation des paramètres est à nouveau lancé et les paramètres estimés sont mis à jours permettant à nouveau de choisir la meilleure vue suivante.

Pour éviter de tomber dans des *minima* locaux qui amènent naturellement la caméra à s'approcher de l'objet, les vues sont acquises sur une sphère de vision centrée sur le centre de la quadrique courante et ayant pour rayon la distance séparant la position courante de la caméra et de l'objet. Seule la composante de translation est calculée, l'orientation sera ajustée pour que le centre de l'objet soit toujours centré dans la vue. Ainsi, le critère calculé ne dépend que de la position de la caméra et de l'estimation courante des paramètres de l'objet. Le problème revient donc à calculer le gradient de $\det(\mathbf{I} + \mathbf{g}_{j+1}\mathbf{G}_j^{-1}\mathbf{g}_{j+1}^\top)$ relatif à la position de la caméra sous la contrainte de rester au voisinage de la position courante et sur une sphère de vision centrée sur l'objet.

Pour résoudre ce problème d'optimisation, nous avons choisi d'utiliser une méthode d'optimisation numérique non linéaire qui ne nécessite pas d'estimation du Jacobien. La méthode considérée est celle du Simplexe de Nelder et Mead [Nelder 65]². Dans un espace de m paramètres, cette méthode consiste à évaluer le poids de la fonction de coût en $m+1$ points particuliers, qui sont les sommets du simplexe, qui est un triangle généralisé dans un espace à m dimensions. Selon la valeur du coût évaluée en ces points, de nouveaux points sont calculés en prenant en compte des considérations géométriques sur la forme du simplexe multidimensionnel. L'algorithme est décrit dans l'encadré 9.1.

Pour reconstruire l'objet au mieux, le déplacement attendu est une rotation autour de l'axe principal de l'objet. Cependant comme la direction de l'axe principal est évaluée à chaque ajout de caméra le parcours obtenu n'est pas parfaitement circulaire.

9.3 Résultats Expérimentaux

Cette section présente les résultats de reconstruction obtenus à partir de données réelles et les améliorations apportées par la vision active en terme de reconstruction et de nombre de vues nécessaires. La figure 9.6 présente les résultats de reconstruction obtenus pour quatre objets sur des arrières-plans quelconques.

L'estimation de la forme des objets par des quadriques donne une information sur l'orientation de l'objet, permet d'affiner l'information sur sa position obtenue lors de l'étape de localisation et donne une estimation de ses dimensions. Mieux les vues sont positionnées, meilleure est la reconstruction. La méthode de vision active proposée permet de faire évoluer l'estimation des paramètres de l'objet plus rapidement que lorsque des vues sont choisies de manière arbitraire sur une sphère de vision.

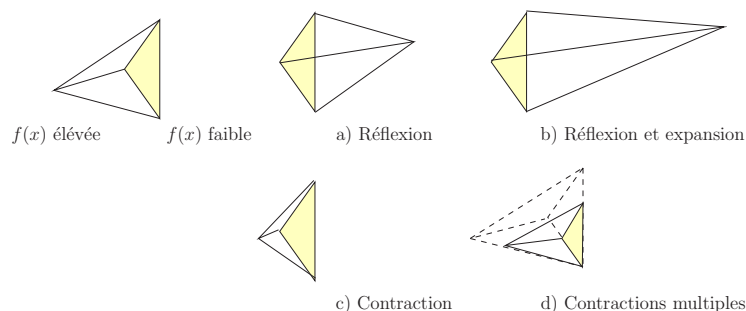
La figure 9.2 illustre la valeur *a priori* de l'entropie pour des positions évaluées sur toute la sphère de vision. On peut remarquer que les valeurs sont réparties de manière symétrique sur la sphère de vision. Le poids de chacune des vues dépend de l'estimation courante de la quadrique et des positions précédemment occupées par la caméra. La symétrie est due aux symétries des quadriques. Il existe deux positions symétrique par rapport au centre de la sphère de vision qui apporteraient des informations identiques. Pour limiter les déplacements de la caméra, c'est la position la plus proche de la position courante qui est choisie comme prochaine meilleure vue. La figure 9.3 présente les déplacements de la caméra et les meilleures vues suivantes calculées.

La figure 9.4 présente l'évolution des paramètres lorsque les vues ajoutées sont sélectionnées de manière arbitraire puis après l'ajout d'une vue sélectionnée par vision active. L'ajout de cette vue bien choisie permet de faire évoluer à nouveau les paramètres. L'ajout ultérieur de vues ne permet plus d'obtenir un meilleur résultat.

Les figures 9.5 et 9.6 présentent les résultats de reconstruction obtenus pour des objets sur fond noir puis sur fond complexe. Dans le cas des fonds complexes, les contours actifs ont été initialisés à la main assez proches des contours réels. Ils sont ensuite suivis dans une séquence de vues.

²Le lecteur intéressé pourra trouver une implémentation de cette méthode dans [Press 07]

Simplexe de Nelder et Mead



Sur un espace de paramètres de dimension 3, le simplexe est un tétraèdre. La fonction de coût est évaluée à chacun de ses sommets. Le point de plus forte valeur est remplacé par un nouveau point qui peut être obtenu par a) réflexion, b) réflexion et expansion, c) contraction sur une direction, ou d) contractions multiples selon toutes les directions et vers le point de poids le plus faible.

1. Choix de $m + 1$ points de l'espace à m dimensions des inconnues, formant un simplexe : x_1, x_2, \dots, x_{m+1} ,
2. Calcul des valeurs de la fonction f en ces points. Les points sont ordonnés et ré-indexés de façon à avoir $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{m+1})$,
3. Calcul de x_0 , centre de gravité de tous les points sauf x_{m+1} ,
4. Calcul de $x_r = x_0 + (x_0 - x_{m+1})$ (réflexion de x_{m+1} par rapport à x_0),
5. - Si $f(x_r) < f(x_1)$, calcul de $x_e = x_0 + 2(x_0 - x_{N+1})$ (étirement du simplexe).
- Si $f(x_e) < f(x_r)$, remplacement de x_{N+1} par x_e , sinon, remplacement de x_{N+1} par x_r et retour à l'étape 2.
6. - Si $f(x_n) < f(x_r)$, calcul de $x_c = x_{N+1} + 1/2(x_0 - x_{N+1})$ (contraction du simplexe).
- Si $f(x_c) \leq f(x_r)$, remplacement de x_{N+1} par x_c et retour à l'étape 2,
7. Similitude de rapport 1/2 et de centre x_1 : remplacement de x_i par $x_0 + 1/2(x_i + x_1)$ et retour à l'étape 2.

FIG. 9.1 – Algorithme de Nelder-Mead.

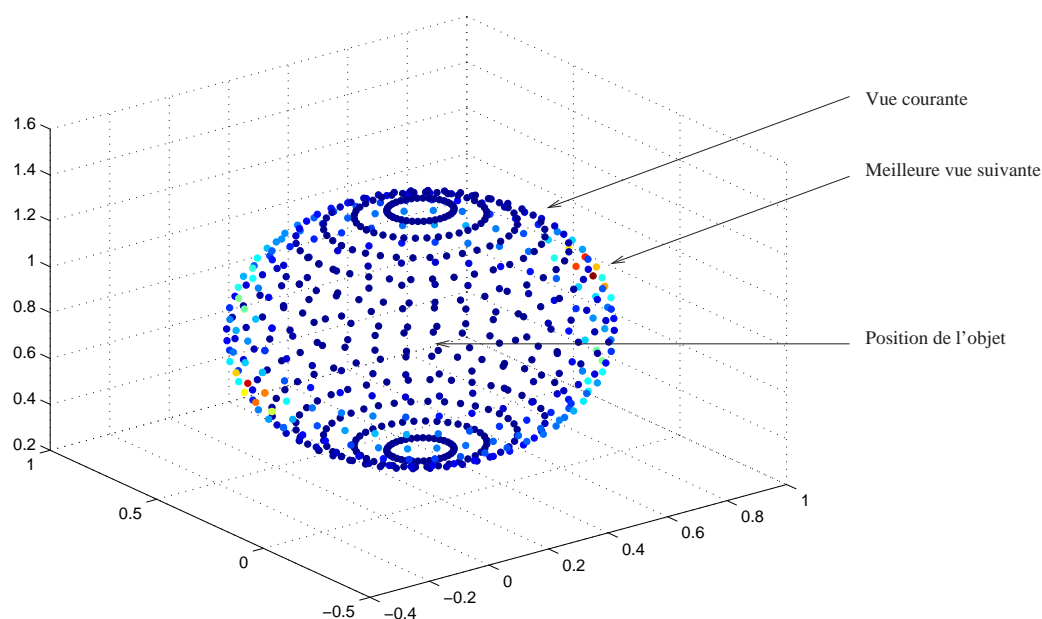


FIG. 9.2 – Estimation de l'incertitude sur les vues suivantes. Les couleurs froides représentent les valeurs faibles du critère à maximiser ($\det(\mathbf{I} + \mathbf{g}_{j+1}\mathbf{G}_j^{-1}\mathbf{g}_{j+1}^\top)$). Les couleurs chaudes représentent de fortes valeurs. La meilleure vue suivante est représentée en rouge sombre.

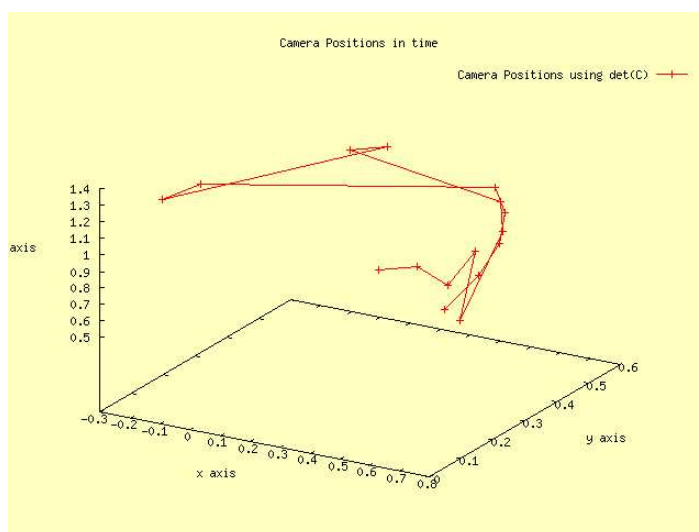


FIG. 9.3 – Déplacement de la caméra sur la sphère de vision. Vues optimales calculées successivement.

Dans un premier temps, trois vues arbitraires sont acquises sur la sphère de vision centrée sur l'objet. Les vues suivantes sont acquises par vision active. Le processus est répété jusqu'à ce que les paramètres estimés de la quadrique n'évoluent plus.

La figure 9.5 présente les résultats de localisation et de caractérisation d'objets placés sur un fond noir. Les coniques estimées dans les vues et les coniques calculées par projection de la quadrique courante se superposent. Sur un arrière-plan noir, le contour actif suit très bien le contour réel de l'objet. La reconstruction est donc très bonne. On peut cependant noter, que dans le cas du premier objet, la poignée a été coupée. Cependant cela ne perturbe pas la caractérisation au sens où nous

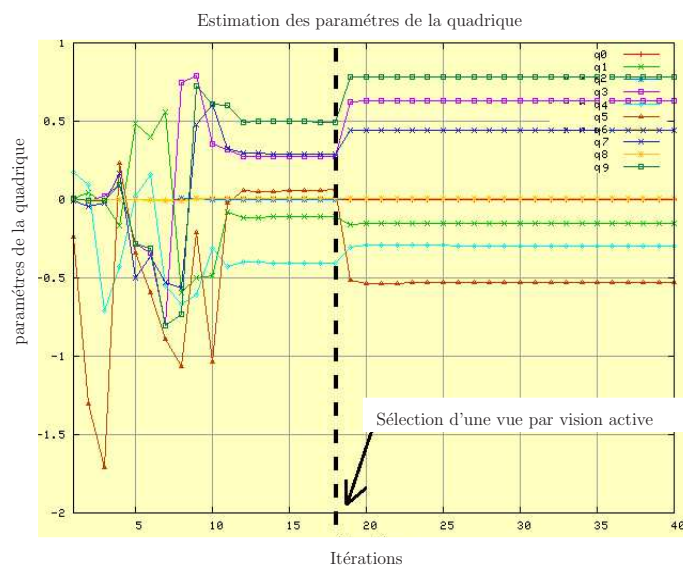


FIG. 9.4 – Évolution des paramètres après l'ajout d'une vue sélectionnée par vision active.

l'avons définie. Notons également qu'une représentation par une quadrique permet d'approximer la forme d'un objet parallélépipédique.

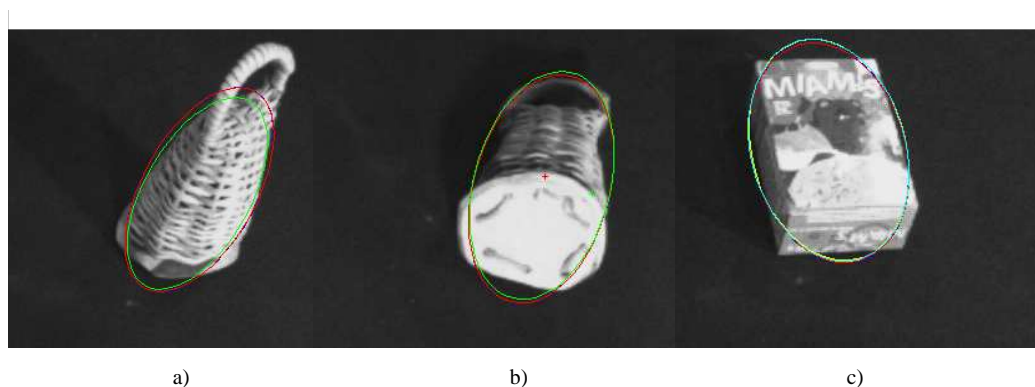


FIG. 9.5 – Estimation des caractéristiques de l'objet.

La figure 9.6 présente quatre résultats de reconstruction. La quadrique estimée courante est projetée dans l'image en jaune, la conique mesurée dans l'image est tracée en rouge et le repère relié à l'objet leur est superposé. Dans ces quatre exemples le grand axe de l'objet est représenté en bleu, le moyen axe en vert et le petit axe en rouge. Les scènes présentent un fond complexe. Tant que le contour actif converge vers le contour réel de l'objet, la quadrique est relativement bien estimée.

La série de vues correspondant à l'image en haut à gauche fait apparaître une très forte réflexion sur l'arrière-plan. Le contour actif n'est pas perturbé et converge vers le contour de l'objet. La caractérisation de l'objet permet de détecter ses axes principaux et ses dimensions le long de ces axes.

En bas à gauche, le décalage entre la projection de la quadrique courante et la quadrique estimée est due à la présence d'une ombre très marquée qui a accroché le contour actif sur plusieurs vues. En haut à droite, l'ombre sous la pomme accroche le contour actif ce qui allonge légèrement la quadrique. Enfin, en bas à droite, la figure présente à la fois la quadrique estimée et en vert la localisation

approximative de l'objet. Dans ces quatre exemple, les repères estimés permettent d'aligner la pince de façon à saisir l'objet.

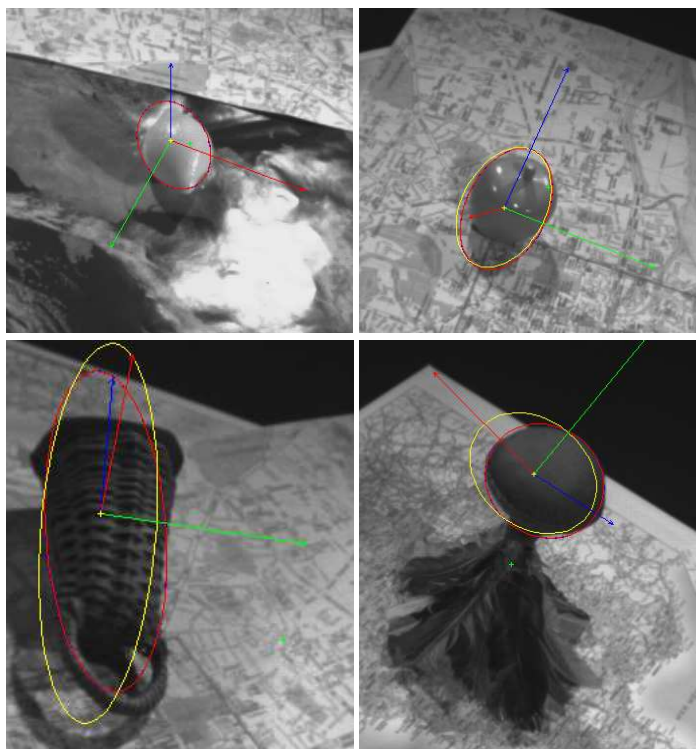


FIG. 9.6 – Estimation des caractéristiques de l'objet.

9.4 Conclusion

Dans ce chapitre, nous avons montré que la méthode de reconstruction proposée donne de bons résultats sur des données réelles. La quadrique estimée affine la localisation de l'objet obtenue à l'issue de la première partie et permet de caractériser des objets de forme globalement convexe en donnant l'orientation de leurs axes principaux et leurs dimensions sur ces axes.

L'utilisation de la vision active permet de sélectionner des vues qui apportent des informations supplémentaires par rapport à celles données par les vues précédemment acquises. En choisissant les nouvelles vues de façon appropriée, la vision active peut faire sortir l'estimation courante d'un minimum local dans lequel elle est tombée pour l'amener à converger vers l'optimum. Seules quelques vues sont nécessaires pour obtenir une bonne estimation des paramètres.

Lorsque l'objet considéré est placé sur un fond noir, l'estimation de ses caractéristiques est très bonne. La projection de la quadrique projetée coïncide avec la conique estimée. Lorsqu'il est placé sur une fond texturé sur lequel des défauts d'illumination apparaissent des petites erreurs peuvent être observées. Ces imprécisions de détection sont dues au fait que le contour actif s'accroche sur les ombres marquées.

La méthode de vision active que nous proposons repose sur les paramètres courants de la quadrique estimée et sur la position des vues acquises précédemment. Elle est donc purement géométrique et ne tient pas compte des conditions d'illumination. Pour éviter que les contours actifs ne soient perturbés par les ombres et les reflets, un positionnement par rapport à l'emplacement estimé de la source lumineuse pourrait être introduit dans le calcul de la vue suivante par vision active.

Conclusion et perspectives

Le travail que nous venons de présenter dans cette thèse propose des outils pour le développement d'une assistance technique à la saisie pour les personnes en situation de handicap. La thèse que nous défendons est qu'un objet peut être saisi par un bras manipulateur guidé par la vision sans disposer d'informations *a priori* et en demandant à l'utilisateur une participation minimale d'un unique *clic* sur une vue de la scène. Nous soutenons également qu'une reconstruction précise de l'objet et de l'environnement n'est pas nécessaire et qu'une localisation et une caractérisation grossière de l'objet suffisent, en supposant que l'objet est globalement convexe et de dimensions adéquates pour être saisi par la pince du manipulateur. Cette thèse peut être vue comme une méthodologie complémentaire aux méthodes existantes reposant sur des bases de données.

Nous avons tout d'abord proposé une solution au problème de la **localisation grossière d'un objet inconnu**. La méthode repose sur l'utilisation d'un système de vision à deux caméras, l'une est déportée et observe la scène, l'autre est embarquée sur l'effecteur du robot. D'après l'information donnée par le *clic* de l'utilisateur dans la vue déportée, on sait que l'objet se trouve sur une droite 3D passant par le centre de la caméra déportée et le *clic*. La caméra embarquée est commandée par asservissement visuel sur la géométrie épipolaire pour visualiser l'ensemble de cette droite. Les informations visuelles extraites des images acquises sont ensuite mises en correspondance avec les informations contenues au voisinage du *clic* pour estimer la position tridimensionnelle de l'objet. La méthode proposée permet tout à la fois :

- une bonne résistance aux différences de points de vue ;
- une prise en compte des incertitudes de mesure ;
- une certaine robustesse aux changements d'illumination ;
- une bonne robustesse aux mouvements relatifs de l'objet par rapport à la caméra déportée.

En fin de processus, l'objet désigné se trouve dans le champ de vision des deux caméras. Ces deux vues peuvent être utilisées pour initier une estimation plus précise de la forme et des dimensions de l'objet suffisante pour la saisie.

Le problème de la **caractérisation de la forme de l'objet** a été traité dans le cadre d'une observation monoculaire dynamique. La forme de l'objet est modélisée par une quadrique dont les paramètres sont estimés à partir des contours détectés dans un ensemble d'images. Cette modélisation donne une information sur la position, l'orientation et les dimensions de l'objet. Ce sont ces informations que nous avons retenues comme étant suffisante à la saisie d'un objet convexe.

Nous avons également proposé une méthode pour segmenter un objet sur un arrière plan. Elle repose sur l'utilisation des contours actifs initialisés à partir de la localisation grossière de l'objet donnée par l'étape précédente. Le contour initial peut être choisi comme une boîte englobante de dimensions *ad hoc* ou comme l'ellipse correspondant aux moments 2D évalués à l'échelle caractéristique.

Le résultat de la reconstruction partielle peut être directement utilisé pour positionner la pince du manipulateur. Cette méthode présente plusieurs avantages :

- une représentation compacte de l'objet ;
- la caractérisation de l'objet peut être affinée en utilisant toutes les images jusqu'à la saisie ;
- une certaine robustesse aux erreurs de calibration.

Nous avons montré que la caractérisation de l'objet est d'autant plus précise que les vues utilisées pour l'estimer sont bien choisies. La dernière contribution de ce mémoire est une méthode de sélection par vision active des vues optimales pour la reconstruction. Compte tenu de l'estimation courante des paramètres de la quadrique qui correspond à la forme de l'objet et des vues précédemment acquises la meilleure vue suivante peut être choisie en recherchant la position de la caméra qui apporte le plus d'information (au sens de Shannon), c'est-à-dire la vue suivante la plus ambiguë. Plus la vue est ambiguë *a priori* plus les informations qu'elle apportera feront évoluer les paramètres estimés.

Le critère à optimiser est l'entropie qui peut être directement calculée à partir du résultat de la reconstruction. Des contraintes supplémentaires peuvent être aisément intégrées dans le processus d'optimisation, par exemple pour interdire certaines positions telles que les positions hors de la zone de travail du bras ou encore pour éviter des obstacles.

Ce travail soulève de nombreuses perspectives de recherche qui n'ont malheureusement pas pu être couvertes dans cette thèse. En premier lieu, une méthode d'analyse quantitative de la précision des résultats de localisation et de caractérisation de l'objet pourrait être mise en place pour évaluer les résultats obtenus. Par exemple, en caractérisant un objet dont le modèle CAO est connu et pourrait être localiser en utilisant les travaux existants dans l'équipe Lagadic [Comport 06].

La redondance d'information donnée par les deux caméras pourrait être mieux exploitée. Par exemple, pour développer une méthode de gestions des occultations et d'évitement d'obstacles. Si l'objet est occulté dans une des deux vues, et si cette occultation peut être détectée alors la seconde pourrait être utilisée pour palier cette situation. D'autre part, la caméra déportée donne une vue globale de la scène qui pourrait être utilisée pour situer à la fois la caméra embarquée et les obstacles qui juchent son parcours. Ainsi, une commande *eye-to-hand* d'évitement des obstacles peut utiliser les degrés de liberté disponible lors de l'exécution des tâche de localisation et de reconstruction.

D'autre part l'évitement des occultations peut être directement intégré comme une contrainte dans le processus de vision active. De la caractérisation d'un objet isolé, nous pourrions alors passer à la caractérisation d'un objet dans une scène complexe.

Le problème rémanent est celui de la détection et de la localisation des obstacles en utilisant la vision et sans information *a priori* sur la scène. Ce point pourrait être résolu en utilisant les points 3D calculés lors de l'étape de localisation pour détecter les obstacles potentiels et les plans composant la scène.

Un autre axe de recherche est de tenter de lever une à une les hypothèses poser sur l'objet ou le système de vision. Dans cette étude, nous avons par exemple limité le type d'objets reconstruits à des objets compacts et globalement convexes. En détectant les inflexions dans les contours 2D, il serait possible de décomposer l'objet en sous parties convexes et approximer ces sous parties par des quadriques pour aller vers une saisie d'objets de forme quelconque.

Finalement L'étalonnage du système est une hypothèse très forte, même si en pratique les méthodes proposées sont relativement robustes à de légères imprécisions du système de mesure. Des méthodes reposant sur des techniques sans étalonnage apporteraient une amélioration notable, en particulier pour que les outils développes puissent être portées sur un robot médical imprécis et flexible comme le bras MANUS.

Rappel des notations

$\mathcal{R}_a(O_a, \mathbf{i}, \mathbf{j}, \mathbf{k})$	repère cartésien orthonormé associé à a
\mathbb{E}^3	espace Euclidien
\mathbb{P}^3	espace Projectif
${}^a\mathcal{X}$	point 3D dans le repère \mathcal{R}_a
${}^a\bar{\mathbf{X}}({}^aX, {}^aY, {}^aZ)$	ses coordonnées Euclidiennes dans le repère \mathcal{R}_a
${}^a\mathbf{X}({}^a\bar{\mathbf{X}}, 1)$	ses coordonnées projective dans le repère \mathcal{R}_a
${}^a\mathbf{M}_b$	matrice homogène de passage de \mathcal{R}_a vers \mathcal{R}_b
${}^b\mathbf{R}_a$	rotation entre \mathcal{R}_a et \mathcal{R}_b
${}^b\mathbf{t}_a$	translation entre \mathcal{R}_a et \mathcal{R}_b
A	est la matrice de projection
K	est la matrice d'étalonnage
${}^a\mathbf{x}$	un point 2D dans le plan image de la caméra a
${}^a\mathbf{x}_m({}^ax, {}^ay, 1)$	les coordonnées métriques homogènes associés
${}^a\mathbf{x}_p({}^au, {}^av, 1)$	les coordonnées pixémiques associés
f	la distance focale
$\mathbf{x}_c(u_0, v_0)^\top$	le point central
(l_x, l_y)	les dimensions d'un pixel
${}^a\Pi$	le plan image de la caméra a
$(O, \mathbf{u}, \mathbf{v})$	le repère associé à la caméra

Le bras MANUS

Dans ce chapitre, on rappelle les spécificités du bras MANUS.

Modèle cinématique

Le Manus est un bras six axes de type RRRRRR, c'est-à-dire six liaisons rotoïdes, à huit degrés de liberté (les six degrés de liberté du positionnement de la pince, un degré d'élévation et l'ouverture de la pince).

Il peut être décomposé en 7 corps C_i , $i = 0..6$, où C_0 est le bâti (cf figure B). $H = 501mm$, $g1 = 400mm$, $g2 = 320mm$ et $g3 = 100mm$ sont respectivement les distances entre les centres des repères O_0 et O_1 , O_2 et O_3 , O_3 et O_4 , O_1 et O_2 .

Modélisation géométrique directe du bras

On peut représenter la situation \mathcal{R}_i du corps C_i par rapport au corps C_{i-1} par les 4 paramètres de la convention de Denavit-Hartenberg modifié [Khalil 86] :

- d_i est la longueur de la perpendiculaire commune aux axes L_i et L_{i-1} , c'est la distance, le long de x_{i-1} , du support de z_{i-1} à z_i ;
- r_i est la distance, le long de z_i , du support de x_{i-1} à x_i ;
- α_i est l'angle entre z_{i-1} et z_i mesuré autour de x_{i-1} ;
- θ_i est l'angle entre x_{i-1} et x_i mesuré autour de z_i .

Soit q_i la position articulaire de l'axe i , les paramètres géométriques du bras Manus sont donc les suivants :

axe :	α	d	θ	r
1	0	0	q_1	H
2	$\pi/2$	0	q_2	$\gamma3$
3	0	$\gamma1$	q_3	0
4	$\pi/2$	0	q_4	$\gamma2$
5	$-\pi/2$	0	q_5	0
6	$\pi/2$	0	q_6	0

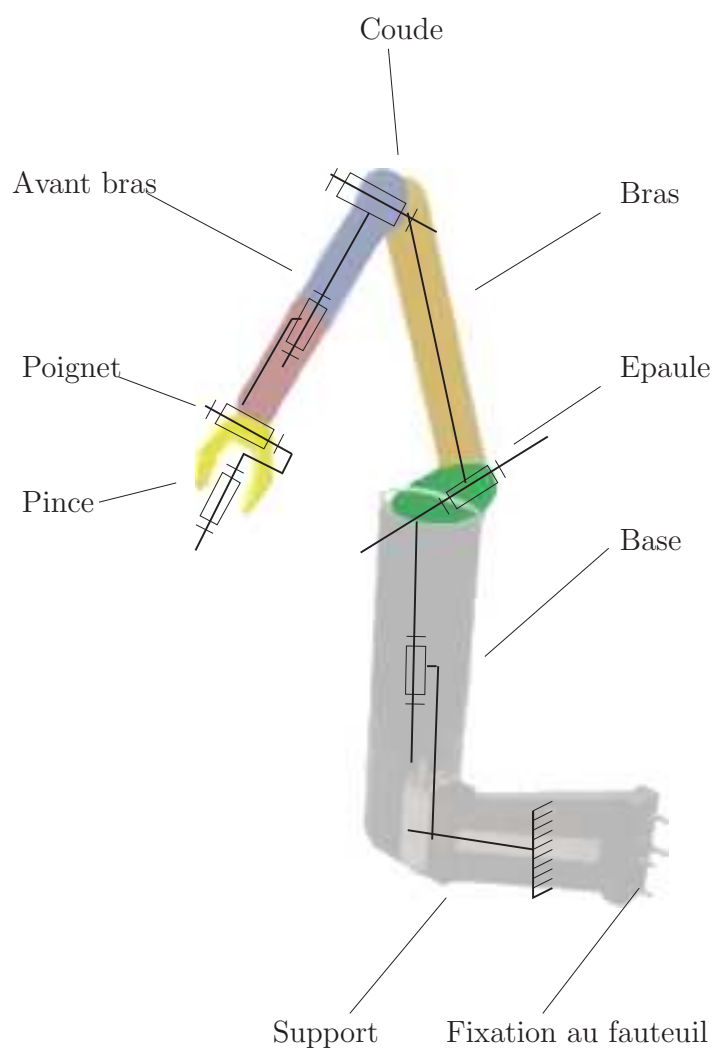


FIG. B.1 – Les six liaisons rotoïdes du bras MANUS

On en déduit les matrices homogènes de passage du repère \mathcal{R}_{i-1} au repère \mathcal{R}_i :

$$\mathbf{T}^{i-1,i} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & d_i \\ \cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i & -r_i \sin \alpha_i \\ \sin \alpha_i \sin \theta_i & \sin \alpha_i \cos \theta_i & \cos \alpha_i & r_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.1})$$

De la position moteur à la position articulaire

Les positions articulaire que l'on évoquait au paragraphe précédent ne sont pas lu directement sur les codeurs places sur les axes. La valeur retournée par les codeurs est ce que l'on appelle la position moteur. Pour transformer cette position en position articulaire il faut tenir compte des couplage mécaniques du système, des positions des repères permettant de définir le modèle géométrique du bras et des décalage dus a la position des zéros de ces repères.

Pour passer de la position moteur à la position articulaire, on doit applique la transformation suivante :

$$\mathbf{q}_a = {}^a \mathbf{C}_m \mathbf{q}_m + {}^a \Delta_m \quad (\text{B.2})$$

${}^a \mathbf{C}_m$ est une matrice carrée de dimension 6 qui représente les couplage des différents axes. La matrice ${}^a \Delta_m$ représente les décalages entre les zéros des mesures sur les codeurs et les zéros des positions articulaires correspondant au modèle géométrique.

Si on considère que le rapport de couplage est de k_i (réel) suivant l'axe i , on obtient alors une matrice de couplage du bras MANUSeST du type :

$${}^a \mathbf{C}_m = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & k_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & k_3 & 1 & k_5 & 0 \\ 0 & 0 & 0 & k_4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.3})$$

Hypothèses

Cette annexe rappelle les hypothèses émises sur les objets et la scène pour parvenir à la saisie d'un objet inconnu en milieu humain à partir de l'information donnée par un seul *clïc* dans l'image de la vue déportée.

Partie 1		
hypothèse 1	Un point de l'objet est sélectionné par l'utilisateur.	1.3.1
hypothèse 2	La scène est statique.	1.3.1
hypothèse 3	Les objets sont génériques.	1.3.1
hypothèse 4	Les objets sont rigides.	1.3.1
hypothèse 5	Les objets sont détectables en utilisant la vision.	1.3.2
hypothèse 6	Les objets sont dans la zone de travail du bras.	1.3.2
hypothèse 7	Les objets sont dans la vue déportée.	1.3.2
hypothèse 8	Le système de vision est étalonné.	2.1.2.2
Partie 2		
hypothèse 9	L'objet est globalement convexe.	5.1
hypothèse 10	L'objet présente un axe de symétrie.	5.1
hypothèse 11	La densité massique de l'objet est uniforme.	5.1
hypothèse 12	Une estimation grossière de la position de l'objet est donnée	5.4
hypothèse 13	Le contour de plus fort contraste est le contour de l'objet.	8
hypothèse 14	L'objet est inclus dans chaque image.	8

Démonstrations

Cette annexe présente les preuves des trois propriétés sur le calcul des valeurs propres données dans la partie D.

Definition

Si λ est une valeur propre de la matrice \mathbf{A} alors, il existe un vecteur \mathbf{x} non nul, telle que $\mathbf{Ax} = \lambda\mathbf{x}$.

Propriété 1.

si λ est une valeur propre de \mathbf{A} , alors $1 + \lambda$ est une valeur propre de $\mathbf{I} + \mathbf{A}$.

Démonstration À partir de la définition, on obtient directement :

$$\begin{aligned}\mathbf{Ax} &= \lambda\mathbf{x} \\ \mathbf{x} + \mathbf{Ax} &= \mathbf{x} + \lambda\mathbf{x} \\ (\mathbf{I} + \mathbf{A})\mathbf{x} &= (1 + \lambda)\mathbf{x}\end{aligned}$$

Propriété 2.

si \mathbf{B} est une matrice telle que $\dim(B) = n \times m$ et \mathbf{A} est une matrice carrée de dimensions $m \times m$ alors les valeurs propres non nulles de $\mathbf{B}^\top\mathbf{BA}$ sont égales aux valeurs propres non nulles de \mathbf{BAB}^\top

Démonstration Pour toute matrice \mathbf{A} de dimension $n \times m$, et toute matrice \mathbf{B} de dimension $m \times n$, avec m et n entier positif tels que $n \geq m$, et en notant $\chi_{\mathbf{AB}}(T)$ le polynôme caractéristique de \mathbf{AB} et $\chi_{\mathbf{BA}}(T)$ le polynôme caractéristique de \mathbf{BA} , on a

$$\chi_{\mathbf{AB}}(T) = (-T)^{n-m}\chi_{\mathbf{BA}}(T)$$

On montre ainsi que les espaces propres pour les valeurs propres non nulles ont même dimension. Et si $n > m$ alors \mathbf{AB} a $n - m$ valeurs propres nulles en plus de celle de \mathbf{BA}

Preuve de la propriété 3. Si \mathbf{A} est une matrice carrée de dimensions $m \times m$ semi définie positive alors pour tout vecteur \mathbf{x} de dimension m non nul, l'inégalité suivante est vérifiée :

$$\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$$

Soit \mathbf{B} , une matrice de dimensions $n \times m$, soit un vecteur \mathbf{y} tel que $\mathbf{y} = \mathbf{B} \mathbf{x}$ alors

$$\mathbf{x}^\top \mathbf{B}^\top \mathbf{A} \mathbf{B} \mathbf{x} \geq 0$$

donc $\mathbf{B}^\top \mathbf{A} \mathbf{B}$ est semi définie positive.

Bibliographie

- [Baker 98] S. Baker, S.K. Nayar. – A theory of catadioptric image formation. – *IEEE Int. Conf. on Computer Vision*, pp. 35–42, 1998.
- [Balaguer 05] C Balaguer, A Gimenez, A Jardon, R Cabas, R Correal. – Live experimentation of the service robot applications for elderly people care in home environments. – *IEEE Int. Conf. on Robotics and Automation*, 2005.
- [Bascle 92] B. Bascle, R. Deriche. – Features extraction using parametric snakes. – *IAPR Int. Conf. on Pattern Recognition*, pp. 659–662, The Hague, Netherlands, 30 Aug-3 Sep 1992.
- [Basri 99] R. Basri, E. Rivlin, I. Shishoni. – Visual homing : Surfing on the epipoles. *Int. Journal of Computer Vision*, 33(2) :117–137, février 1999.
- [Batavia 91] A.I. Batavia, G. DeJong, L.B. McKnew. – Toward a national personal assistance program : The independent living model of long-term care for persons with disabilities. *Journal of Health Politics, Policy and Law*, 16(3) :523–545, 1991.
- [Beaton 74] A.E. Beaton, J.W. Tukey. – The fitting power series, meaning polynomials, illustrated on band spectroscopic data. – *Tecnometrics*, vol. 16, pp. 147–185, 1974.
- [Becker 98] M. Becker, E. Kefalea, E. Mael, C. Von der Malsburg, M. Pagel, J. Triesch, Wurtz R.P. Vorbruggen, J.C., S. Zadel. – Gripsee : a robot for visually guided grasping. – *ICANN*, 1998.
- [Becker 99] M. Becker, E. Kefalea, E. Maël, M. von der Malsburg, C. and Pagel, J. Triesch, J.C. Vorbrüggen, R. P. Würtz, S. Zadel. – GripSee : A gesture-controlled robot for object perception and manipulation. *Autonomous Robots*, 6(2) :203–221, 1999.
- [Beis 97] J.F. Beis, D.G. Lowe. – Shape indexing using approximate nearest neighbour search in high-dimensional spaces. *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 1000–1006, 1997.
- [Benlamri 04] R. Benlamri, Y. Al-Marzooqi. – Free-form object segmentation and representation from registered range and color images. *Image and Vision Computing*, 22(9) :703–717, march 2004.
- [Blake 95] A. Blake. – A symmetry theory of planar grasp. *Int. Journal of Robotics Research*, pp. 425–444, octobre 1995.
- [Bonnas 97] J.F. Bonnas, J.C. Gilbert, C. Lemaréchal, C. Sagastizabal. – *Optimisation numérique, aspects théoriques et pratiques*. – Springer, 1997.
- [Boukir 93] S. Boukir. – *Reconstruction 3D d'un environnement statique par vision active*. – PhD. Thesis, Université de Rennes 1, IRISA, octobre 1993.

- [Bouteloup 94] V. Bouteloup. – *Reconstruction 3D d'objets complexes par vision active*. – Rapport de DEA, Université de Rennes 1, 1994.
- [Brouard 04] C. Brouard, N. Dutheil, P. Gilbert, H. Michaudon, S. Vanovermeir, P. Tisserand, C. Vaslin, M. Maudinet, A. Piquet, J. Sanchez. – Le handicap en chiffres, Février 2004.
- [Busnel 01] M. Busnel, R. Gelin, Lesigne B. – Evaluation of a robotized master/raid workstation at home : protocol and first results. – *IEEE Int. Conf. on Rehabilitation Robotics*, 2001.
- [Casals 93] A. Casals, R. Villa, D. Casals. – A soft assistance arm for tetraplegics. – *1st TIDE Cong.*, pp. 103–107, avril 1993.
- [Casselès 97] V. Casselès, R. Kimmel, G. Sapiro. – Geodesic active contours. *Int. Journal of Computer Vision*, pp. 61–79, 1997.
- [Chaumette 90a] F. Chaumette. – *La relation vision-commande : théorie et application à des tâches robotiques*. – PhD. Thesis, Université de Rennes 1, juillet 1990.
- [Chaumette 90b] F. Chaumette. – *La relation vision-commande : théorie et application à des tâches robotiques*. – PhD. Thesis, Université de Rennes 1, IRISA, July 1990.
- [Chaumette 00] F. Chaumette, E. Marchand. – A new redundancy-based iterative scheme for avoiding joint limits : application to visual servoing. – *IEEE Int. Conf. on Robotics and Automation, ICRA '00*, vol. 2, pp. 1720–1725, San Francisco, USA, April 2000.
- [Chaumette 06] F. Chaumette, S. Hutchinson. – Visual servo control, part i : Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4) :82–90, December 2006.
- [Chaumette 07] F. Chaumette, S. Hutchinson. – Visual servo control, part ii : Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1) :109–118, March 2007.
- [Chezi 06] G. Chezi, G. L. Mariottini, D. Prattichizzo. – Epipole-based visual servoing for mobile robots. *Advance Robotics*, 20(2) :255–280, 2006.
- [Chung 96] R. Chung, K-K. Ho. – Using 2d active contour models for 3d reconstruction from serial sections. – *ICPR '96 : Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I*, p. 849, Washington, DC, USA, 1996. IEEE Computer Society.
- [Cohen 93] L. Cohen, I. Cohen. – Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11) :1131–1147, November 1993.
- [Colle 02] E. Colle, Y. Rybarczyk, Hoppenot. P. – Arph : An assistant robot for disabled people. – *SMC'02*, 2002.
- [Colle 08] E. Colle, P. Hoppenot. – Arph : Handicap08. – *HANDICAP'08*, 2008.
- [Collewet 09] C. Collewet. – Polar snakes : a fast and robust parametric active contour model. – *soumis à IEEE Int. Conf. on Image Processing*, Le Caire, Egypte, Novembre 2009.
- [Comport 05] A.I. Comport, E. Marchand, F. Chaumette. – Efficient model-based tracking for robot vision. *Advanced Robotics, Special issue on "Selected papers from IROS'04"*, 19(10) :1097–1113, October 2005.
- [Comport 06] A.I. Comport, E. Marchand, M. Pressigout, F. Chaumette. – Real-time markerless tracking for augmented reality : the virtual visual servoing framework. *IEEE Trans. on Visualization and Computer Graphics*, 12(4) :615–628, July 2006.
- [Coudert 97] F. Coudert. – *REconstruction 3D des surfaces courbes modélisées par des quadriques*. – Rapport de DEA, Université de Rennes 1, 1997.

- [Cross 98] G. Cross, A. Zisserman. – Quadric surface reconstruction from dual-space geometry. – *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, pp. 25–31, janvier 1998.
- [Curtis 08] N. Curtis, J. Xiao. – Efficient and effective grasping of novel objects through learning and adapting with a knowledge base. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'08*, pp. 2252–2257, Nice, France, September 2008.
- [Dallaway 95] J.L. Dallaway, R.D. Jackson, P.H.A. Timmers. – Rehabilitation robotics in europe. *IEEE Trans. on Rehab. Eng.*, pp. 35–45, 1995.
- [Davison 03] A.J. Davison. – Real-time simultaneous localisation and mapping with a single camera. – *International Conference on Computer Vision, ICCV 2003*, Nice, France, june 2003.
- [Dollar 06] A.M. Dollar, R.D. Howe. – A robust compliant grasper via shape deposition manufacturing. *IEEE/ASME Transactions on Mechatronics*, 11(2) :154–161, 2006.
- [Dune 07a] C. Dune, C. Leroux, E. Marchand. – Intuitive human interactive with an arm robot for severely handicapped people - a one click approach. – *IEEE Int. Conf. on Rehabilitation Robotics, ICORR'2007*, pp. 582–589, Noordwijk, The Netherlands, June 2007.
- [Dune 07b] C. Dune, E. Marchand, C. Leroux. – Coopération multi-caméras pour la recherche puis le positionnement par rapport à un objet. – Congrès francophone des jeunes chercheurs en vision par ordinateur, ORASIS'07, June 2007.
- [Dune 07c] C. Dune, E. Marchand, C. Leroux. – One click focus with eye-in-hand/eye-to-hand cooperation. – *IEEE Int. Conf. on Robotics and Automation*, Roma, 2007.
- [Dune 07d] C. Dune, E. Marchand, C. Leroux. – Reconstruction active de surfaces implicites basée contours. – Journées Nationales de la Recherche en Robotique, JNRR'07, (session poster), October 2007.
- [Dune 08a] C. Dune, E. Marchand, C. Collewet, C. Leroux. – Active rough shape estimation of unknown objects. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'08*, Nice, France, September 2008.
- [Dune 08b] C. Dune, A. Remazeilles, E. Marchand, C. Leroux. – Vision-based grasping of unknown objects to improve disabled people autonomy. – *Robotics : Science and Systems Manipulation Workshop : Intelligence in Human Environments*, Zurich, Switzerland, June 2008.
- [Edsinger 06] A. Edsinger, Kemp C. C. – Manipulation in human environments. – *IEEE/RSJ International Conference on Humanoid Robotics*, pp. 102–109, Genoa, Italy 06, 2006.
- [Eftring 99] H. Eftring, K. Boschian. – Technical results from manus user trials. – *IEEE Int. Conf. on Rehabilitation Robotics*, pp. 136–141, 99.
- [Elena 03] M. Elena, M Critiano, F. Damiano, M. Bonfe. – Variable structure pid controler for cooperative eye-in-hand/eye-to-hand visual servoing. – *IEEE. Int. Conf. on Control Applications, ICCA'03*, pp. 989–994, Istanbul, Turkey, 2003.
- [ElKhoury 07] S. El Khoury, A Sahbani, V. Perdereau. – Learning the natural grasping component of an unknown object. – *IEEE Int. Conf. on Robotics and Automation*, pp. 2957–2962, San Diego, CA, october 2007.
- [Espiau 92] B. Espiau, F. Chaumette, P. Rives. – A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3) :313–326, juin 1992.
- [Faugeras 93] O.D. Faugeras. – *Three dimensional Computer Vision : A Geometric Viewpoint*. – MIT Press, 1993.

- [Flandin 00] G. Flandin, F. Chaumette, E. Marchand. – Eye-in-hand / eye-to-hand cooperation for visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, pp. 2741–2746, San Francisco, CA, avril 2000.
- [Flandin 02] G. Flandin, F. Chaumette. – Visual data fusion for objects localization by active vision. – *Eur. Conf. on Computer Vision, ECCV'02, LNCS 2353*, pp. 312–326, Copenhagen, Denmark, May 2002.
- [Franco 06] J.-S. Franco, E. Boyer. – Visual shapes of silhouette sets. – *In Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization and Transmission*, 2006.
- [Gangloff 98] J. Gangloff, M. de Mathelin, G. Abba. – 6 dof high speed dynamic visual servoing using gpc controllers. – *IEEE Int. Conf. on Robotics and Automation, ICRA '98*, pp. 2008–2013, mai 1998.
- [Gardin 94] J. Gardin, T. Lindeberg. – Direct computation of shape cues using scale-adapted spatial derivative operators. *Int. Journal of Computer Vision*, 1994.
- [Gates 07] B. Gates. – A robot in every home. – *Scientific American Magazine*, january 2007.
- [Glover 08] J. Glover, D. Rus, N. Roy. – Probabilistic models of object geometry for grasp planning. – *Proceedings of Robotics : Science and Systems IV*, Zurich, Switzerland, June 2008.
- [Graf 04] B. Graf, M. Hans, R.D. Schraft. – Care-o-bot ii :development of a next generation robotic home assistant. *Autonomous robots*, 2004.
- [Harris 88] C. Harris, M. Stephens. – A combined corner and edge detector. – *Alvey Vision Conference*, pp. 147–151, 1988.
- [Hartley 97] R. Hartley. – In defence of the 8-points algorithm. – *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 580–593, 1997.
- [Hartley 01] R. Hartley, A. Zisserman. – *Multiple View Geometry in Computer Vision*. – Cambridge University Press, 2001.
- [Harwin 95] W. S. Harwin, T. Rhaman, R. A. Foulds. – A review of design issues in rehabilitation robotics with reference to north american research. *IEEE Transaction on rehabilitation engineering*, 3(1), march 1995.
- [Hashimoto 93a] K. Hashimoto (édité par). – *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. – World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapor, 1993.
- [Hashimoto 93b] K. Hashimoto, H. Kimura. – Dynamic visual servoing with nonlinear model-based control. – *Proceedings of the 12th World Congress IFAC*, vol. 9, pp. 405–408, Sidney, Autralia, juillet 1993.
- [Hauck 99] A. Hauck, M. Rüttinger, J.and Sorg, G. Färber. – Visual determination of 3d grasping points on unknown objects with a binocular camera system. – *IEEE Int. Conf. on Robotics and Automation*, pp. 272 – 278., 1999.
- [Hillman 03] M. Hillman. – Rehabilitation robotics from past to present - a historical perspective. – *Int. Conf. on Rehabilitation Robotics*, avril 2003.
- [Horaud 06] R. Horaud, Knossow D., M. Michaelis. – Camera cooperation for achieving visual attention. *Machine Vision and Applications*, 16(6) :1–12, 2006.
- [Huebner 08a] K. Huebner, D. Kragic. – Selection of robot pre-grasp using box-based shape approximation. – *IEEE Int. Conf. on Robotics and Automation*, pp. 1765–1770, Nice, France, 22-26 septembre 2008.

- [Huebner 08b] K. Huebner, S. Ruthotto, D. Kragic. – Minimum volume bounding box decomposition for shape approximation in robot grasping. – *IEEE Int. Conf. on Robotics and Automation*, pp. 1628–1633, Pasadena, CA, 2008.
- [Iossifidis 02] Ioannis Iossifidis, Carsten Bruckhoff, Christoph Theis, Claudia Grote, Gregor Schöner. – Cora : An anthropomorphic robot assistant for human environment. – *Proceedings of the 11th IEEE International Workshop On Robot and Human Interactive Communication, IEEE ROMAN 2002*, Berlin, Germany, 2002. IEEE.
- [Ishii 95] S. Ishii, S. Tanaka, F. Hiramatsu. – Meal assistance robot for severely handicapped people. – *IEEE Int. Conf on Rehabilitation and Automation*, pp. 1308–1313, San Francisco, USA, 1995.
- [Jones 99] T. Jones. – Raid : Toward greater independence in the office& home environment. – *IEEE Int. Conf. on Rehabilitation Robotics*, pp. 201–206, 1999.
- [Karl 94] W.C. Karl, G.C. Verghese, A.S. Willsky. – Reconstructiong ellipsoids from projections. – *CVGIP : Graphical Models and Image Processing*, vol. 56, pp. 124–139, 1994.
- [Kass 87] M. Kass, A Wtikin, D. Terzopoulos. – Snakes : active contour models. *Int. Journal of Computer Vision*, 4 :321–331, 1987.
- [Katz 08] D. Katz, J. Kenney, O. Brock. – How can robots succeed in unstructured environments? – *Robotics : Science and Systems IV Manipulation Workshop : Intelligence in Human Environments*, Zurich, Switzerland, June 2008.
- [Kawamura 94a] K. Kawamura, M. Iskarous. – Trends in service robots for the disabled and the elderly. – Special session on service robots for the disabled and elderly people, 1994.
- [Kawamura 94b] S. Kawamura, K.and Bagchi, M. Iskarous, R. T. Pack, A. Saad. – Intelligent robotic aid system for human services. – *AIAA/NASA Conf. Intelligent Robotics Fields*, vol. 2 of *Factory, Service Space*, pp. 413–420, march 1994.
- [Kayikcioglu 00] T. Kayikcioglu, A. Gangal, M. Ozer. – Reconstructing ellipsoids from three projection contours. *Pattern Recognition Letters*, 21 :959–968, 2000.
- [Kazi 95] Z. Kazi, M. Salganicoff, M. Beitler, S. Chen, D. Chester, R. Foulds. – Multimodal user supervised interface and intelligent control (musiic. – *In AAAI 1995 Fall Symposium Series on Embodied Language and Action MIT*, pp. 47–58, 1995.
- [Kemp 07] C.C. Kemp, A. Edsinger, E. Torres-Jara. – Challenges for robot manipulation in human environments. *IEEE Robotics and Automation Magazine*, pp. 20–29, march 2007.
- [Kutulakos 99] K.N. Kutulakos, S.M. Seitz. – A theory of shape by space carving. *International Journal of Computer Vision*, 38 :307–314, 1999.
- [Kwee 89] H.H. Kwee, J.J. Duimel, J.J. Smits, A.A. Tuinhof de Moed, J.A. van Woerden. – The manus wheelchair-borne manipulator : System review and first results. – *IARP, 2nd Workshop Medical and Healthcare Robotics*, pp. 385–395, 1989.
- [Lane 96] J.P. Lane, D.J. Usiak, J.A. Moffatt. – Consumer criteria for assistive devices :operationalizing generic criteria for specific abledata categories. – *Rehabilitation Engineering and Assistive Technology Society of North America*, 1996.
- [LeCam 08] M. Le Cam. – *Segmentation d’objets pour la désignation en un clic*. – Rapport de DEA, Telecom Bretagne, 2008.
- [Leroux 04] C Leroux, M Guerrand, C. Leroy, Y Méasson, B. Boukarri. – Magritte : a graphic supervisor for remote handling interventions. – *ESA Workshop on Advanced Space Technologies for Robotics and Automation, ’ASTRA 2004*, Noordwijk, The Netherlands, novembre 2004.

- [Leroux 06] C. Leroux, G. Chalubert, O. Tahri, S. Schmutz, N. Biard, I. Lafont, J-F. Diert, J.-M. Alexandre, R. Gelin. – Interface intelligente pour la saisie d’objets robotisée, handicap 2006. – *National Conf. Handicap*, paris, june 2006.
- [Leroux 07] C. Leroux, I. Laffont, N. Biard, S. Schmutz, J.F. Desert, G. Chalubert, Y. Meason. – Robot grasping of unknown objects, description and validation of the function with quadraplegic people. – *IEEE Int. Conf. on Rehabilitation Robotics, ICORR’2007*, pp. 35–42, Noordwijk, The Netherlands, June 2007.
- [Li 08] W.H. Li, M. Zhang, L. Kleeman. – Bilateral symmetry detection for real-time robotics applications. *Int. Journal of Robotics Research*, 27 :785–813, september 2008.
- [Liefhebber 07] F. Liefhebber, J. Sijs. – Vision-based control of the manus using sift. – *IEEE Int. Conf. on Rehabilitation Robotics, ICORR’2007*, pp. 854–861, Noordwijk, The Netherlands, June 2007.
- [Liegois 77] A. Liegeois. – Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. on Systems, Man and Cybernetics*, 7(12) :868–871, décembre 1977.
- [Lindeberg 98] T. Lindeberg. – Feature detection with automatic scale selection. *Int. Journal of Computer Vision*, 30(2) :79–116, november 1998.
- [Lippiello 05] V. Lippiello, B. Siciliano, L. Villani. – Eye-in-hand/eye-to-hand multi-camera visual servoing. – *IEEE Int. Conf. on Decision and Control, CDC’05*, pp. 5354 – 5359, Seville, Spain, décembre 2005.
- [Lippiello 07] V. Lippiello, B. Siciliano, L. Villani. – Position based visual servoing in industrial multirobot cells using a hybrid camera configuration. *IEEE Trans. on Robotics*, pp. 73 – 86, 2007.
- [LonguetHiggins 81] Longuet-Higgins. – A computer algorithm for reconstructing a scene from two projections. *Nature*, 293 :133–135, 1981.
- [LopezNicolas 06] G. Lopez-Nicolas, C. Sagues, J.J. Guerrero, D. Kragic, P. Jensfelt. – Nonholonomic epipolar visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, pp. 2378–2384, may 2006.
- [Lowe 04] D.G. Lowe. – Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2) :91–110, 2004.
- [Ma 94] S.D. Ma, X. Chen. – Quadric surface reconstruction from its occluding contours. – *IAPR Int. Conf. on Pattern Recognition*, 1994.
- [Ma 96] S.D. Ma, L. Li. – Ellipsoid reconstruction from three perspective views. – *IAPR Int. Conf. on Pattern Recognition*, pp. 344–348, 1996.
- [Mahoney 01] R. Mahoney. – The raptor wheelchair robot system. – *IEEE Int. Conf. on Rehabilitation Robotics*, pp. 135–141, Evry, France, 2001.
- [Malis 05] E. Malis, E. Marchand. – Méthodes robustes d’estimation pour la vision robotique. – *jnrr*, 2005.
- [Malladi 95] R. Malladi, J. Sethian, B Vemuri. – Shape modelling with front propagation : A level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(2) :158–175, 1995.
- [Mallat 99] S. Mallat. – *A Wavelet Tour of Signal Processing*. – Academic Press, 1999.
- [Mann 05] W.C. Mann. – *Smart technology for aging, disability, and independance, the state of the science*. – Wiley Interscience, 2005.
- [Mansard 07] N. Mansard, F. Chaumette. – Task sequencing for sensor-based control. *IEEE Trans. on Robotics*, 23(1) :60–72, February 2007.

- [Marchand 96] E. Marchand, F. Chaumette, A. Rizzo. – Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'96*, vol. 3, pp. 1083–1090, Osaka, Japan, novembre 1996.
- [Marchand 04] E. Marchand. – *Contributions à la commande d'une caméra réelle ou virtuelle dans des mondes réels ou virtuels*. – Rennes, Habilitation à diriger les recherches, Université de Rennes, novembre 2004.
- [Mariottini 04] G. L. Mariottini, D. Prattichizzo, G. Oriolo. – Epipole based visual servoing for nonholonomic mobile robots. – *IEEE Int. Conf. on Robotics and Automation*, pp. 2378–2384, may 2004.
- [Mariottini 05] G.L. Mariottini, E. Alunno, J. Piazzzi, D. Prattichizzo. – Epipole-based visual servoing with central catadioptric camera. – *IEEE Int. Conf. on Robotics and Automation*, pp. 3516–3521, 2005.
- [Mariottini 07] G. L. Mariottini, G. Oriolo, D. Prattichizzo. – Image-based visual servoing for nonholonomic mobile robot using epipolar geometry. *IEEE Trans. on Robotics*, 23(1) :87–100, february 2007.
- [Marotta 02] A. Marotta, J. Piazzzi, A. Prattichizzo, D. and Vicino. – Epipole based 3d visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 366– 371, october 2002.
- [Marquardt 63] D.W. Marquardt. – An algorithm for least-square estimation of non-linear parameters. *Journal of the Society of Industrial and Applied Mathematics*, 1963.
- [Menet 90] S. Menet, P. Saint-Marc, G. Medioni. – Active contour models : Overview, implementation and applications. – *Int. Conf. Systems, Man, and Cybernetics*, vol. 212, pp. 194–199, 1990.
- [Michel 06] C. Michel, V. Perdereau, M. Drouin. – An approach to extract natural grasping axes with a real 3d vision system. – *IEEE International Symposium on Industrial Electronics*, vol. 4, pp. 3130–3135, July 2006.
- [Mikolajczyk 04] K. Mikolajczyk, C. Schmid. – Scale and affine invariant interest point detectors. *Int. Journal of Computer Vision*, 60 :63–86, 2004.
- [Mikolajczyk 05] K. Mikolajczyk, C. Schmid. – A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27 :1615–1630, 2005.
- [Mikolajczyk 06] K. Mikolajczyk, C. Schmid. – Indexed based on scale invariant interest points. – *IEEE Int. Conf. on Computer Vision*, vol. 4, pp. 525–531, July 2006.
- [Muis 05] A. Muis, K. Ohnishi. – Eye-to-hand approach on eye-in-hand configuration within real-time visual servoing. *IEEE/ASME Transactions on Mechatronics*, 10(4) :404–410, 2005.
- [Nadeau 08] C. Nadeau. – *Reconstruction volumique d'objet pour un saisie robotique autonome*. – Rapport de DEA, ENSEEIHT, 2008.
- [Nadeau 09] C. Nadeau, C. Dune. – Vers une saisie en un seul clic : caractérisation de la forme 3d d'un objet à partir d'informations visuelles. – *Soumis à Congrès francophone des jeunes chercheurs en vision par ordinateur, ORASIS'09*, 2009.
- [NaitChabane 06] K. Nait-Chabane. – *Exploitation de la redondance pour la commande coordonnée d'un manipulateur mobile d'assistance aux personnes handicapées*. – PhD. Thesis, Université d'Evry, novembre 2006.
- [Nelder 65] J. A. Nelder, R. Mead. – A simplex method for function minimization. *Computer Journal*, 7 :308–313, 1965.
- [Nelson 93] B. Nelson, P.K. Khosla. – Increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance. – *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 418–423, Atlanta, Géorgie, mai 1993.

- [Nelson 94a] B. Nelson, P.K. Khosla. – Integrating sensor placement and visual tracking strategies. – *IEEE Int. Conf. Robotics and Automation*, vol. 2, pp. 1351–1356, San Diego, mai 1994.
- [Nelson 94b] B. Nelson, P.K. Khosla. – The resolvability ellipsoid for visual servoing. – *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pp. 829–832, Seattle, Washington, juin 1994.
- [Nguyen 08] H. Nguyen, C. Anderson, A. Trevor, A. Jain, Z. Xu, C.C. Kemp. – El-e : An assistive robot that fetches objects from flat surfaces. – *Human-Robot Interaction Workshop*, 2008.
- [Norman 90] Norman. – The design of everyday things. – New York Doubleday, 1990.
- [Odobez 95] J.M. Odobez, P. Bouthemy. – Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6 :348–365, 1995.
- [Peters 06] R.A. Peters, R.E. Bodenheimer, O.C. Jenkins. – Sensory-motor manifold structure induced by task outcome : Experiments with robonaut. – *IEEE-RAS International Conference on Humanoid Robot*, pp. 484–489, december 2006.
- [Piazzi 03] J. Piazzi, D. Prattichizzo. – Auto-epipolar strategy for mobile robot visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1802–1807, octobre 2003.
- [Piazzi 04] J. Piazzi, D. Prattichizzo, N. J. Cowan. – Auto epipolar visual servoing. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 363–368, Sendai, Japon, octobre 2004.
- [Precioso 05] F. Precioso, M. Barlaud, T. Blu, M Unser. – Robuste real time segmentation of images and video using a smoothing-spline snake-based algorithm. *IEEE Trans. on Image Processing*, 2005.
- [Press 07] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. – *Numerical Recipes : The Art of Scientific Computing*. – Cambridge University Press, Third Edition (20), 2007.
- [Prock 98] A.C. Prock, C.R. Dyer. – Towards real-time voxel coloring. – *In Proc. DARPA Image Understanding Workshop*, pp. 315–321, 1998.
- [Remazeilles 08] A. Remazeilles, C. Leroux, G. Chalubert. – Sam : a robotic butler for handicapped people. – *IEEE Int. Symp. on Robot and Human Interactive Communication, RO-MAN'08*, pp. 000–0000, août 2008.
- [Rives 00] P. Rives. – Visual servoing based on epipolar geometry. – *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 602–607, Takamatsu, Japan, novembre 2000.
- [Rousel 02] P. Rousel. – Une estimation de la diffusion des aides techniques à partir de l'enquete hid de l'insee. – *Handicap*, n96, p54, 2002.
- [Sahbahni 07] A. Sahbahni, J.-P. Saut, V. Perdereau. – An efficient algorithm for dexterous manipulation planning. – *IEEE International Multi-Conference on Systems, Signals and Devices*, 2007.
- [Samson 91] C. Samson, M. Le Borgne, B. Espiau. – *Robot Control : the Task Function Approach*. – Clarendon Press, Oxford, Royaume Uni, 1991.
- [Sanz 05] P.J. Sanz, A. Requena, J.M. Inesta, A.P. del Pobil. – Grasping the not-so-obvious : vision-based object handling for industrial applications. *IEEE Robotics and Automation Magazine*, 12(3) :44–52, September 2005.

- [Saut 07] J.-P. Saut, A. Sahbani, S. El Khoury, V. Perdereau. – Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07)*, pp. 2907–2912, San Diego, CA, USA, 2007.
- [Saxena 08] A. Saxena, J. Driemeyer, N. Andrew. – Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27 :157–173, 2008.
- [Shashua 97] A. Shashua, S. Toelg. – The quadric reference surface : Theory and applications. *IEEE Int. Conf. on Computer Vision*, 23(2), 1997.
- [Shibuya 08] N. Shibuya, Y. Shimohata, T. Harada, Y. Kuniyoshi. – Smart extraction of desired object from color-distance image with user's tiny scribble. – *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'08*, Nice, France, September 2008.
- [Sian 07] E.N. Sian, K. Yokoi, K. Kajita, S. and Tanie. – Whole-body motion generation integrating operator intention and robot autonomy in controlling humanoid robots. *IEEE Transactions on Robotics*, 23 :763 – 775, August 2007.
- [Siciliano 91] B. Siciliano, J.J. Slotine. – A general framework for managing multiple tasks in highly redundant robotic systems. – *IEEE International Conference on Advanced Robotics (ICAR '91)*, pp. 1211–1216, Pisa, Italy, Juin 1991.
- [Song 98] W.K. Song, H. Lee, J.S. Kim, Y.S. Yoon, Z. Bien. – Kares : Intelligent rehabilitation robotic system for the disabled and the elderly. – *Int. Conf. in Medicine and Biology society*, vol. 20. IEEE, 1998.
- [Soyama 03] R. Soyama, S. Ishii, A. Fukase. – The development of meal-assistance robot 'myspoon'. – *IEEE Int. Conf. on Rehabilitation Robotics*, pp. 88–91, 2003.
- [Speth 08] J. Speth, A. Morales, Sanz. – Vision-based grasp planning of 3d objects by extending 2d contour based algorithms. – *IEEE Int. Conf. on Robotics and Automation*, pp. 2240–2245, Nice, France, septembre 2008.
- [Steven 97] S.M. Steven, C.R. Dyer. – Photorealistic scene reconstruction by voxel coloring. – *International Journal of Computer Vision*, pp. 1067–1073, 1997.
- [Stout 08] B.J. Stout, K.J. Hagglund, M.J. Clark. – The challenge of financing and delivering personal assistant services. *Journal of Disability Policy Studies*, 19(1) :44–51, June 2008.
- [Stranger 94] C.A. Stranger, C. Anglin, W. S. Harwin Harwin, D.P. Romilly. – Devices for assisting manipulation : A summary of user task priorities. *IEEE Transactions on rehabilitation Engineering*, pp. 256–265, December 1994.
- [Sundarrao 06] S. Sundarrao, R. Alqasemi, E. McCaffrey, R. Dubey. – Selection considerations and a comparison of commercial wheelchair mounted robotic arms. – *Florida Conference on Recent Advances in Robotics*, Miami, Florid, May 2006.
- [Taylor 02] G.R. Taylor, L. Kleeman, Wernersson A. – Robust colour and range sensing for robotic applications using a stereoscopic light stripe scanner. – *IEEE Int. Conf. on Robotics and Automation*, pp. 86–91, 2002.
- [Terzopoulos 91] D. Terzopoulos, D. Metaxas. – Dynamic 3d models with local and global deformations : deformable superquadrics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13 :703–713, May 1991.
- [Tikhonov 63] A.N. Tikhonov. – Solution of incorrectly formulated problems and the regularization method. *Soviet Math Dokl English translation of Dokl Akad Nauk SSSR* 151, 4 :501–504, 1963.
- [Topping 98] M. Topping, J. Smith. – The developpement of handy 1, a rehabilitation robotic system to assist the severely disabled. – *Industrial robot*, pp. 316–320, 1998.

- [Tsai 89] R. Tsai, R. Lenz. – A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. on Robotics and Automation*, 5(3) :345–358, juin 1989.
- [Tsui 08] K. M. Tsui, H. Yanco, D. Feil-Seifer, M. Mataric. – Survey of domain-specific performance measures in assistive robotic technology. – Workshop on Performance Metrics for Intelligence Systems (PerMIS), August 2008.
- [VanderLoos 95] H.F.M. Van der Loos. – Va/stanford rehabilitation robotics research and development program : Lessons learned in the application of robotics technology to the field of rehabilitation. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 3 :46–55, mars 1995.
- [VanderLoos 00] H.F.M. Van der Loos, J.J. Wagner, N. Smaby, K Chang, O Madrigal, L J Leifer, O Khatib. – Provar assistive robot system architecture. – *IEEE Int. Conf. on Robotics and Automation*, pp. 741–746, Detroit, mai 2000.
- [VanderLoos 08] H.F.M. Van der Loos, D.J. Reinkenensmeyer. – *Handbook of Robotics*, chap. 53. Rehabilitation and health care Robotics, pp. 1223–1251. – Springer, 2008.
- [Volosyak 05a] I. Volosyak, O. Kouzmitcheva, Ristic D., Graser A. – Improvement of visual perceptual capabilities by feedback structures for robotic system friend. *IEEE Tans. on system, man and cybernetics*, 35 :66–74, février 2005.
- [Volosyak 05b] I Volosyak, Ivlev O., Graser A. – rehabilitation robot friend ii - the general concept and current implementation. – *IEEE Int. Conf. on Rehabilitation Robotics*, pp. 540–544, Chicago, IL, USA, juin 2005.
- [Wang 05] B. Wang, L. Jiang, J.W. Li, Cai H.G., H. Lui. – Grasping unknown objects based on 3d model reconstruction. – *IEEE/ASME International Conference on Advanced Intellingent Mechatronics*, pp. 461–466, Monterey, California, USA, 24-26 Juillet 2005.
- [Whaite 91] P. Whaite, F. Ferrie. – From uncertainty to visual exploration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1038–1049, 1991.
- [Whaite 97a] P. Whaite. – *A curious Machine for Autonomous visual exploration*. – PhD. Thesis, Departement of Electrical Engineering, McGill University, Montréal, August 1997.
- [Whaite 97b] P. Whaite, F. Ferrie. – Autonomous exploration : Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 :339–346, 1997.
- [Xu 98] C. Xu, J. L. Prince. – Snakes, shapes and gradient vector flow. – *IEEE transactions on image processing*, vol. 7, march 1998.
- [Zhang 97] Z. Zhang. – Parameter estimation techniques : a tutorial with application to conic fitting. *image and vision computing*, 1997.

Table des figures

1	Une vision de la robotique personnelle [Gates 07] : plusieurs robots spécialisés sont affectés à des tâches spécifiques dans une maison domotisée qui coordonne l'ensemble.	6
1.1	La robotique médicale. Trois familles de systèmes robotiques peuvent être distinguées, dans le domaine de la chirurgie, de la réhabilitation et de la simulation d'acte médicaux. Parmi les robots de réhabilitation, on trouve les prothèses, les orthèses (ou exosquelettes), les robots d'assistance à la saisie ou à la marche et les robots de rééducation. La limite entre ces catégories est floue et il est courant de considérer que les orthèses et les prothèses sont des robots d'assistance.	11
1.2	À gauche, une prothèse de bras, à droite, une prothèse de cheville et de pied.	12
1.3	À gauche, l'orthèse ARmin de l'ETHZ à droite l'orthèse Able du CEA-LIST. Ces orthèses sont utilisées pour la rééducation du membre supérieur.	12
1.4	Le robot Lokomat rééduque la marche.	13
1.5	Extrait de la brochure <i>Le handicap en chiffres</i> [Brouard 04].	15
1.6	La station de travail Raid (robotic assistance in daily Living) [Jones 99]	20
1.7	Le projet DeVAR/ProVAR [VanderLoos 95, VanderLoos 00]	20
1.8	Le robot Handy	21
1.9	Le robot my spoon	21
1.10	Le robot Tou	22
1.11	Les bras Manus et Raptor	22
1.12	Le bras Manus est un bras sécurisé qui peut être commandé en utilisant un joystick	23
1.13	Le robot Mats peut se déplacer de prise en prise à l'intérieur d'une maison, se fixer sur un support fixe ou encore sur un fauteuil roulant	23
1.14	Le robot fixe ISAC	24
1.15	La station fixe CORA	25
1.16	Le robot Friend et son système de localisation d'objet basé sur la vision et un plateau tactile	26
1.17	Le projet allemand CARE-O-BOT utilise à la fois un laser et une caméra pour détecter les objets et estimer leur forme	26
1.18	Le robot HRP-2 dispose d'une base de données des modèles CAO de son environnement. Le scénario consiste à détecter un réfrigérateur, à l'ouvrir et à saisir une bouteille de soda à l'intérieur. Le système connaît <i>a priori</i> les modèles de la canette et du réfrigérateur.	27
1.19	Le détecteur de points de saisie est entraîné sur des données simulées. Il est équipé d'une caméra qui lui permet de retrouver ces points de saisie sur les images courantes. Le système saisit ensuite les objets en boucle ouverte [Saxena 08].	29
1.20	Un rayon laser balaye la scène et est détecté dans les images d'une caméra [Taylor 02].	29
1.21	La main SDM permet de saisir une grande variété d'objets par enroulement [Dollar 06].	30
1.22	Saisie d'objets en connaissant partiellement leur forme [Saxena 08].	30

1.23	Aviso en cours d'évaluation au centre de Garches. Aviso est un bras MANUS équipé d'un banc de stéréovision et d'une interface homme machine intuitive	33
1.24	Le scénario de saisie AVISO	34
1.25	Coopération caméra déportée - caméra embarquée : les disques verts, rouges et noirs représentent les différentes situations rencontrées en fonction de la position de l'objet à saisir, c'est-à-dire s'il est vu respectivement par les deux caméras, par la caméra déportée uniquement ou s'il n'est visible par aucune des caméras.	37
1.26	Le projet El-E : à gauche, l'objet à saisir est pointé par un laser vert. Le robot détecte le point vert résultant dans l'image d'une caméra omnidirectionnelle. Au milieu, le robot oriente sa tête stéréovision en direction du point détecté et estime la position 3D du point. À droite, le robot va saisir l'objet [Nguyen 08].	38
1.27	La baguette xwand contient une centrale inertielle et un laser. Elle permet de pointer un périphérique d'une maison intelligente et d'agir sur son état : allumer la télévision, allumer ou éteindre une lampe, etc	39
1.28	Le scénario de saisie <i>One Click</i>	40
2.1	Petite histoire du sténopé	50
2.2	Modélé sténopé ou trou d'épingle de la caméra	51
2.3	Projection centrale et notations	51
2.4	Passage des coordonnées métriques aux coordonnées pixeliques	52
2.5	La géométrie épipolaire de deux vues. Le couple de points homologues est ${}^1\mathbf{x}_m$ et ${}^2\mathbf{x}_m$. Les centres des caméras sont O_1 et O_2 . Les épipoles sont e_1 and e_2 dans le plan image de la caméra 1 et dans celui de la caméra 2, respectivement. La droite l_1 est la droite épipolaire associée à ${}^2\mathbf{x}_m$. La droite l_2 est la droite épipolaire associée à ${}^1\mathbf{x}_m$	55
2.6	Définition des repères	58
2.7	Définition de la ligne de vue, issue du point cliqué par l'utilisateur	59
2.8	Stratégie de localisation : la caméra embarquée est déplacée de sorte à acquérir des vues du segment 3D qui porte l'objet d'intérêt.	60
3.1	Les propriétés de la configuration auto épipolaire permettent d'aligner les caméras.	62
3.2	Dispositif d'asservissement visuel sur 4 points ; à gauche, un montage caméra embarquée, dit <i>eye-in-hand</i> ; à droite, un montage caméra déportée, dit <i>eye-to-hand</i>	63
3.3	Schéma bloc simplifié d'un asservissement visuel indirect	64
3.4	à gauche : il existe une infinité de solutions, toute situées sur le noyau \mathbf{J}_1^\top de \mathbf{J}_1 . La pseudo inverse \mathbf{J}_1^+ de \mathbf{J}_1 fournit la solution de norme minimale ; à droite : Dans le cas où une tâche secondaire \mathbf{e}_2 est considérée, elle est projetée sur le noyau \mathbf{J}_1^\top de la tâche et n'a donc aucun effet sur la tâche principale.	66
3.5	Représentation des droites 2D $\mathbf{s} = (\rho, \theta)$	68
3.6	Selon la position relative des deux caméras, lorsque la tâche de centrage d'une extrémité du segment est régulée, la seconde extrémité peut être positionnée derrière le plan image de la caméra embarquée. La matrice d'interaction associée au second point n'est pas valide. La solution est d'effectuer n tâches de centrage successives pour garantir que les points 3D virtuels soient toujours devant le plan image.	70
3.7	évolution de la fonction de coût en fonction de la position articulaire.	71
3.8	Algorithme d'évitement de butée.	71
3.9	À gauche : le robot cartésien de l'équipe Lagadic à l'INRIA Rennes. À droite le robot 6 axes Staubli RX90, du LTC au CEA-LIST	73

3.10	Asservissement visuel sur la ligne épipolaire : En haut à gauche, la vue de la caméra déportée. Un point du masque est sélectionné par l'utilisateur. Les deux autres images du haut représentent les différentes positions occupées par la caméra au cours du parcours de la ligne de vue. Les poses représentées en bleu correspondent à la phase de centrage de la ligne de vue. Les poses acquises en rouge, correspondent au parcours de la ligne de vue à proprement parler. En bas, les trois premières images sont acquises au cours du centrage et les trois images suivantes, au long du parcours de la ligne de vue. La ligne épipolaire associée au <i>clac</i> apparaît en vert, la ligne rouge est la consigne de centrage.	74
3.11	Asservissement visuel sur la ligne épipolaire : en haut, les erreurs des tâches primaire (rouge) et secondaire (vert) ; au milieu, les vitesses de translation ; en bas, les vitesses de rotation. On peut noter que la tâche secondaire engendre principalement un mouvement de rotation Ω_x autour de l'axe \mathbf{j} . En effet, lorsque la tâche primaire est régulée, l'axe \mathbf{j} est normal au plan épipolaire. Une rotation autour de cette axe est donc suffisante pour parcourir la ligne de vue passant par l'objet.	75
3.12	Asservissement visuel sur la ligne épipolaire : En haut à gauche, la vue de la caméra déportée. Un point du dinosaure est sélectionné par l'utilisateur. Les deux autres images du haut représentent les différentes positions occupées par la caméra au cours du parcours de la ligne de vue. Les poses représentées en bleu correspondent à la phase de centrage de la ligne de vue. Les poses acquises en rouge, correspondent au parcours de la ligne de vue à proprement parler. En bas, les trois premières images sont acquises au cours du centrage et les trois images suivantes, au long du parcours de la ligne de vue. La ligne épipolaire associée au <i>clac</i> apparaît en vert, la ligne rouge est la consigne de centrage.	76
3.13	Le bras entre en butée au bout de 58 itérations. La tâche cesse alors de converger. La commande en vitesse reste constante. En réalité ces vitesses ne sont pas appliquées au système qui est en butée. La convergence est interrompue.	78
3.14	Asservissement visuel sur la ligne épipolaire : en haut, les erreurs des tâches primaires (rouge) et secondaire (vert) ; en bas, les vitesses articulaires. Les discontinuités dans la commande sont due au centrage du i points successifs tout au long du segment. Elles peuvent être lissées en faisant varier la position du point courant continuellement plutôt qu'utiliser un ensemble de points discrets.	79
3.15	Images obtenue lors de l'envoi de la commande décrite par la figure 3.14. En rouge les caméra avant l'évitement de butées et en bleu les poses après l'évitement de butée . . .	80
3.16	Suivi d'un objet en mouvement. Évolution des erreurs et des vitesses cartésiennes. Le déplacement de l'objet perturbe la décroissance exponentielle des erreurs. Les tâches sont régulées à une erreur de trainage près.	82
3.17	Suivi d'un objet en mouvement. En haut à gauche l'image déportée initiale où l'objet est sélectionné. En haut à droite le parcours de la caméra embarquée. La deuxième ligne contient les vues successives de la caméra déportée. La troisième ligne le motif suivi dans ces vues. Enfin la dernière ligne présente les vues acquises depuis la caméra embarquée.	83
3.18	Si la caméra embarquée passe entre la caméra déportée et l'objet, alors le motif avoisinant le <i>clac</i> ne peut plus être suivi.	83
4.1	Images acquises par les deux caméras au cours du parcours de la ligne de vue associée au <i>clac</i> . En haut, les images acquises par la caméra déportée qui contiennent l'objet. En bas la vue de la caméra embarquée dans laquelle l'objet se projette par intermittence.	86
4.2	Algorithme des SIFT	88

4.3	Détection des extremums dans l'espace d'échelle : une pyramide gaussienne est construite, puis pour chaque dimension, l'image est convoluée par des filtres gaussien de variance croissante. Les images résultantes sont soustraites deux à deux (DoG) donnant une approximation du Laplacien. Ensuite, chaque pixel est comparé aux huit pixels voisins à la même échelle et aux 9 pixels voisins dans les échelles adjacentes. Si sa valeur est maximum ou minimum, alors il est conservé comme candidat potentiel.	89
4.4	À gauche une image acquise depuis la caméra déportée, à droite, une image acquise depuis la caméra embarquée. Deux paires de points mis en correspondance sont représentées (${}^e\mathbf{x}_1; {}^d\mathbf{x}_1$) et (${}^e\mathbf{x}_2; {}^d\mathbf{x}_2$). La première paire vérifie la contrainte épipolaire tandis que la seconde paire est un faux positif	91
4.5	Localisation de l'objet le long de la ligne de vue. Dans cet exemple, trois vues embarquées sont acquises. L'objet est inclus dans les deux premières vues et absent de la troisième vue. Il est localisé à partir des mises en correspondance entre chacune de ces vues embarquées et la vue déportée.	92
4.6	Algorithme de localisation d'un objet inconnu à partir d'un <i>clik</i>	93
4.7	À partir d'une paire de points homologues et connaissant la position relative des deux caméras, le point 3D ${}^d\mathbf{X}$ correspondant peut être calculé. Sa projection orthogonale ${}^d\mathbf{H}$ sur la ligne de vue associée au point cliqué donne une indication sur la profondeur D de ce point le long de la ligne de vue. La distance h est la distance euclidienne entre le point et la ligne de vue.	94
4.8	Construction de la fonction gaussienne attachée à un point. Son amplitude A est fonction de sa distance à la ligne de vue. Sa variance σ^2 représente l'incertitude sur la position. Elle est liée à sa distance au centre de projection \mathcal{O}_d , à la disparité D de la mesure des points homologues dans l'image et à l'erreur de mesure de la position relative des deux caméras.	95
4.9	La distance h est la distance séparant le point de la ligne de vue (voir 4.7).	96
4.10	Plus le point 3D est distant des centres de projection des caméras, plus une erreur de mesure entraîne une erreur de localisation importante, plus la variance attachée au point est importante.	97
4.11	Incertitude en fonction de la configuration des caméras et de la position du point 3D. Les cercles jaunes représentent les variances associées à la caméra C_{e1} , les cercles rouges sont associés à C_{e2} , les cercles verts à C_{e3} , les cercles de couleur cyan à C_{e4} et bleus à C_{e5}	99
4.12	Mélange de gaussiennes sur la portion de la ligne de vue visible (en vert). En rouge les gaussiennes associées à chacun des points P_i	100
4.13	Acquisition de trois vues successives dans lesquelles apparaissent les mises en correspondance de paires identiques. La normalisation du mélange de gaussiennes sur ces trois vues permet de ne pas surévaluer la contribution d'une même paire de points détectée plusieurs fois. Dans cet exemple, si les résultats ne sont pas normés, la localisation de l'objet est fautive car deux faux positifs sont détectés dans les trois vues.	102
4.14	Détection et mise en correspondance de points SIFT au cours du parcours de la ligne de vue.	104
4.15	Localisation d'une bouteille d'eau sur un fond texturé sur le RX90.	105
4.16	Localisation d'une bouteille d'eau sur un fond uniforme en utilisant le RX90.	105
4.17	Localisation d'un jouet dans une scène complexe.	105
4.18	Plusieurs objets sont tour à tour sélectionnés par un <i>clik</i> , puis localisés et finalement centrés dans l'image de la caméra embarquée.	107

4.19	Plusieurs objets sont tour à tour électionnés <i>clac</i> , localisés et centrés dans l'image de la caméra embarquée. La colonne de gauche présente les images acquise par la caméra embarquée au moment de la sélection de l'objet suivant dans la vue déportée. La seconde colonne présente les fonction de visibilité et la troisième colonne présente l'estimation de la vraisemblance de la position de l'objet. Les courbes vertes correspondent à l'évaluation des fonctions après l'acquisition d'une première vue, les bleues après deux vues et les rouges après trois vues.	108
4.20	Cas limite de l'algorithme de localisation : plusieurs objets semblable à l'objet d'intérêt se trouve sur la ligne de vue.	109
5.1	Objets rigides, globalement convexes et présentant des symétries : a) objets cylindriques (bouteilles, cannettes, conserves, etc.) b) objets parallélépipédiques (boites, téléphone, etc.)	115
5.2	Deux objets présentant des prises adaptées à une saisie humaine. Ces prises ne sont pas nécessairement les mieux adaptées à une saisie robotique, en particulier lorsque la pince possède des mors trop volumineux pour être glissés dans l'anse.	116
5.3	Stratégie de saisie choisie pour des objets génériques et globalement convexes. La pince est alignée avec les axes principaux de l'objet et centrée sur son centre de gravité. En rouge l'axe principal de la pince, en bleu l'axe moyen et en vert le petit axe.	117
5.4	Si les objets sont globalement convexes et présentent uniquement quelques concavités suffisamment petites devant les dimensions de la pince, alors ces concavités ne perturbent pas la saisie. À gauche, les objets sont dans une pose favorable à la saisie, à droite, les pattes des jouets risquent de gêner l'approche de la pince.	117
6.1	Quelques exemple de quadriques de dimension 3 : a) cylindre elliptique, b) ellipsoïde, c) paraboloides elliptiques, d) hyperboloides.	122
6.2	Axes principaux d'une quadrique par ordre décroissant de taille : rouge, vert puis bleu. a) ellipsoïde $\mathcal{Q} = (4, 2, 1, 0, 0, 0, 0, 0, -1)$ b) hyperboloïde à une nappe $\mathcal{Q} = (1, 1, -1, 0, 0, 0, 0, 0, -1)$ c) cylindre $\mathcal{Q} = (1, 1, 0, 0, 0, 0, 0, 0, -1)$ d) hyperboloïde à deux nappes $\mathcal{Q} = (-1, -1, 1, 0, 0, 0, 0, 0, -1)$	124
6.3	Projection perspective d'un ellipsoïde.	125
6.4	Exemple de projection d'une quadrique sur trois vues. À gauche, conique projetée à partir d'un ellipsoïde. À droite, coniques projetées à partir d'un hyperboloïde à une nappe.	128
7.1	Estimation des paramètres de la quadrique	130
7.2	Distance algébrique ($\propto (b^2/c^2 - 1)$) et euclidienne (d) d'un point \mathbf{x} à une conique \mathcal{C}	132
7.3	Reconstruction duale.	134
7.4	Algorithme de Levenberg Marquardt.[Marquardt 63]	137
7.5	Algorithme des moindres carrés pondérés itérés IRLS (Iteratively Reweighted Least Squares).	139
7.6	Simulation non bruitée : une quadrique \mathcal{Q}_r (en rouge) est observée depuis six points de vues (pyramides) répartis sur une sphère de vision. Le système est considéré non bruité. a) Vue du dessus. b) Vue de côté c) Projections dans les plans images. Les points jaunes sont les projections des points de l'objet, les cercles rouges sont les points de contours détectés et les ellipses bleues sont les ellipses estimées.	140
7.7	Simulation non bruitée : résultats de reconstruction d'une quadrique à partir de ses projections en utilisant l'espace dual. a) Vue de dessus, b) Vue de côté, c) Projection dans les vues. La quadrique estimée se projette parfaitement (en vert) sur les coniques estimées (points rouges).	141

7.8	Projection de la quadriques \mathcal{Q}_r dans 6 vues. a) Aucun bruit n'est ajouté. Dans les deux images suivante, les positions des points de contours de la première vue sont bruitées par un bruit additif gaussien a) de variance 0.01 et de moyenne 0.02 b) de variance 0.1 et de moyenne 0.2. Dans le second cas, l'estimation de la conique dans la première vue apporte une information totalement fausse.	142
7.9	a) caractéristiques de l'objet : en bleu le résultat de la méthode duale, en noir celui de la méthode non linéaire et en vert celui de la méthode robuste b) reconstruction duale c) reconstruction non linéaire, d) reconstruction robuste. Les trois figures horizontales sont les projections de la quadrique reconstruite par la méthode e) duale, f) non linéaire g) robuste	143
7.10	Simulation faiblement bruitée, reconstruction par la méthode robuste. a) vue de dessus de la quadrique estimée b) vue de cote de la vue estimée c) coniques projetées en rouge et coniques estimées en vert. d) évolution des poids associés à chaque vue e) évolution des paramètres. Après 10 itérations, le poids associé à la première vue chute à zéro. . .	144
7.11	a) Caractérisations obtenues : en bleu par la méthode duale, en noir par la méthode non linéaire et en vert par la méthode robuste. b) reconstruction duale c) reconstruction non linéaire. d) reconstruction robuste. Les trois figures horizontales sont les projections de la quadrique reconstruite par la méthode e) duale, f) non linéaire g) robuste, dans chacune des vues.	145
7.12	Simulation fortement bruitée, reconstruction par la méthode robuste. a) Caractéristiques déterminées par la méthode robuste b) vue de côté de la quadrique estimée c) estimation des coniques projetées. d) Évolution des poids. e) Évolution des paramètres.	147
7.13	Vues proches : une quadrique \mathcal{Q}_r (en rouge) est observée depuis six points de vues (pyramides) proches. a) vue du dessus, b) vue de côté c) projections non bruitée dans les plans images d) projection bruitée dans les plans images. Les points jaunes sont les projections des points de l'objet, les cercles rouges sont les points de contours détecté et les ellipses bleues sont les ellipses estimées.	148
7.14	a) objet réel b) reconstruction duale c) reconstruction non linéaire, d) reconstruction robuste. Les trois figures horizontales sont les projections de la quadrique reconstruite par la méthode e) duale, f) non linéaire g) robuste	149
7.15	a) objet réel b) reconstruction duale c) reconstruction non linéaire, d) reconstruction robuste. Les trois figures horizontales sont les projections de la quadrique reconstruite par la méthode e) duale, f) non linéaire g) robuste, dans chaque vue.	150
7.16	Un parallélépipède est observé depuis six vues. a) vue de dessus b) vue de côté c) projections dans les six vues	151
7.17	Estimation de la quadrique qui correspond au mieux à un objet de forme parallélépipédique qui est observé depuis six vues. a))Les caractéristiques estimées par les différentes méthodes : en bleu la méthode duale, en noir la méthode non linéaire, en vert la méthode robuste. Reconstructions obtenues : b) par la méthode duale, c) par la méthode non linéaire c) par la méthode robuste. Projections associées à d) la méthode duale e) la méthode non linéaire f) la méthode robuste.	152
7.18	Un objet de forme quelconque est observé depuis six vues. a) vue de dessus b) vue de côté c) projections dans les six vues	153
7.19	Objet de forme quelconque a) ses caractéristiques estimées par les différentes méthodes : en bleu la méthode duale, en noir la méthode non linéaire, en vert la méthode robuste. Reconstructions obtenues : b) par la méthode duale, c) par la méthode non linéaire d) par la méthode robuste. Projections associées à e) la méthode duale f) la méthode non linéaire g) la méthode robuste.	154
8.1	Segmentation à partir de la couleur d'un objet et détection des contours associée. Les objets sont de couleurs uniformes et chaque objet a une couleur discriminante [Glover 08].	156

8.2	Comme dans le cadre de notre étude, l'objet est désigné par pointage. La segmentation repose sur l'hypothèse que l'objet se trouve à l'emplacement de la zone la plus texturée de l'image. À gauche l'image à traiter, au centre les contours détectés et à droite la détection du blob contenant l'objet.	156
8.3	Segmentation par recherche des symétries [Li 08].	157
8.4	Segmentation reposant sur une carte de disparité et la détection de région d'une même couleur [LeCam 08].	157
8.5	Contour actif : l'évolution des contours est soumise à des forces externes (dues à l'image) et internes (dues au contour). Dans la version classique des contours actifs [Kass 87], chaque point se déplace en fonction des forces qui lui sont appliquées (flèches rouges).	158
8.6	Analogie contour actif - système physique : le contour peut être assimilé à un système physique de topologie similaire plongé dans un fluide et soumis à des champs forces, par exemple, la gravité et les forces de frottement fluide.	159
8.7	Détection de contours par contour actif. Les contours sont initialisés par la sélection de quatre points autour de l'objet. Le processus de détection converge au bout d'une cinquantaine d'itérations. Le contour initial est tracé en jaune. L'évolution du contour correspond à l'ensemble de courbes bleues, le contour à convergence est une courbe verte.	162
8.8	Suivi du contour dans une séquence d'images. Dans la première image, le contour est initialisé par quatre points et converge vers le contour réel de l'objet. Dans les images suivantes de la séquence, les contours sont initialisés en utilisant le contour détecté dans l'image précédente. La méthode de suivi est robuste à des déplacements relativement faible de l'objet entre deux vues.	163
8.9	Suivi d'un objet de couleur uniforme sur une arrière plan complexe. Le contour actif est initialisé à l'intérieur de l'objet, par un petit rectangle centré sur le <i>clic</i> . Le centre du contour est mis à jour après convergence et est utilisé comme point d'initialisation pour la détection du contour dans l'image suivante de la séquence. Dans l'image 1) le contour jaune est le contour initial, les contours bleus sont les positions intermédiaires et le contour vert est le contour à convergence.	165
8.10	Deux exemples d'initialisation à partir d'un rectangle centré sur le <i>clic</i> et de côté la distance au bord le plus proche. Les objets sont placés sur un fond uniforme. Le rectangle jaune est le contour initial, les courbes bleues sont les contours intermédiaires et le contour vert est le contour à la convergence.	166
8.11	Suivi d'un objet sur un fond uniforme. Le contour actif est initialisé à l'extérieur de l'objet, par une boîte englobante centrée sur un point appartenant à l'objet et de côté, deux fois la distance séparant ce point des bords de l'image. Après convergence, le point central du contour est mis à jour en calculant le barycentre du contour et la boîte englobante correspondant. Cette nouvelle boîte est utilisée comme initialisation pour l'image suivante.	167
8.12	Détection de l'échelle d'un seul objet en évaluant le maximum du LoG.	168
8.13	Détection de l'échelle d'un seul objet en évaluant le maximum du LoG.	168
8.14	Détection de l'échelle d'objets composant une scène.	169
8.15	Évaluation de trois détecteurs d'échelle, la norme du laplacien (a) et le déterminant (b) et la trace (c) de la matrice des moments 2D pour différentes échelles.	170
8.16	Estimation de l'échelle et de l'orientation d'une boîte	171
8.17	Estimation de l'échelle et de l'orientation pour différents objets. Pour tous les objets testés, un cercle bleu est centré sur le point sélectionné. Son rayon est l'échelle caractéristique en ce point. L'ellipse rouge et ses deux axes vert et jaune sont calculés à partir de la matrice des moment 2D évaluée au point sélectionné et à l'échelle caractéristique.	171
8.18	La valeur de l'échelle caractéristique dépend de la position du point pour laquelle elle est calculée.	172
8.19	deux exemples.	172

9.1	Algorithme de Nelder-Mead.	180
9.2	Estimation de l'incertitude sur les vues suivantes. Les couleurs froides représentent les valeurs faibles du critère à maximiser ($\det(\mathbf{I} + \mathbf{g}_{j+1} \mathbf{G}_j^{-1} \mathbf{g}_{j+1}^\top)$). Les couleurs chaudes représentent de fortes valeurs. La meilleure vue suivante est représentée en rouge sombre. 181	181
9.3	Déplacement de la caméra sur la sphère de vision. Vues optimales calculées successivement. 181	181
9.4	Évolution des paramètres après l'ajout d'une vue sélectionnée par vision active.	182
9.5	Estimation des caractéristiques de l'objet.	182
9.6	Estimation des caractéristiques de l'objet.	183
B.1	Les six liaisons rotoïdes du bras MANUS	190

Localisation et caractérisation d'objets inconnus à partir d'informations visuelles : vers une saisie intuitive pour les personnes en situation de handicap

Le point de départ des travaux présentés dans cette thèse est la volonté de développement d'une aide robotisée à la saisie intuitive pour les personnes en situation de handicap. L'outil proposé est un manipulateur contrôlé en utilisant directement les informations transmises par deux caméras, l'une embarquée sur la pince qui donne une vue détaillée de la scène, et l'autre déportée qui en offre une vue d'ensemble. L'objet de nos travaux est de saisir un objet *a priori* inconnu à partir d'un seul *clac* de l'utilisateur sur une image acquise par la caméra déportée. Nous proposons des méthodes permettant de localiser et de caractériser grossièrement un objet de forme convexe afin qu'il puisse être saisi par la pince du manipulateur. Cette thèse peut être vue comme complémentaire aux méthodes existantes reposant sur l'utilisation de bases de données. Ce manuscrit est divisé en deux parties : la localisation grossière d'un objet inconnu et la caractérisation de sa forme.

L'objet se situe sur la ligne de vue qui passe par le centre optique de la caméra déportée et le *clac*. La projection de cette ligne de vue dans la caméra embarquée est la ligne épipolaire associée aux *clac*. Nous proposons donc un asservissement visuel reposant sur l'utilisation de la géométrie épipolaire pour commander la caméra embarquée de façon à parcourir cette ligne. Les indices visuels extraits des images embarquées sont ensuite mis en correspondance avec les indices détectés au voisinage du *clac* pour estimer la position 3D de l'objet. Cette méthode est robuste à des mouvements relatifs de l'objet et de la caméra déportée au cours du processus de localisation. En fin de processus, l'objet désigné se trouve dans le champ de vision des deux caméras et ces deux vues peuvent servir à initier une caractérisation plus précise de l'objet et suffisante pour la saisie.

Le problème de la caractérisation de la forme de l'objet a été traité dans le cadre d'une observation monoculaire dynamique. La forme de l'objet est modélisée par une quadrique dont les paramètres sont estimés à partir de ses projections dans un ensemble d'images. Les contours de l'objet sont détectés par une méthode de contours actifs initialisés à partir de la localisation grossière de l'objet. La caractérisation de l'objet est d'autant plus précise que les vues utilisées pour l'estimer sont bien choisies. La dernière contribution de ce mémoire est une méthode de sélection par vision active des vues optimales pour la reconstruction. Les meilleures vues sont choisies en recherchant les positions de la caméra qui maximisent l'information.

Mots-clés : Robotique d'assistance, Asservissement visuel, Modélisation 3D, Vision Active

Vision-based localisation and characterisation of unknown objects : towards an intuitive grasping tool for the disabled

The starting point of this study is the development of a robot assistant for the disabled. The robot is a vision-based controlled manipulator which is equipped with two cameras : one is embedded on the gripper and gives a close view of the scene while the second one is remotely located and gives a global view of the scene. The objective is then to grasp an *a priori* unknown object given the only information of one *click* on the remote image. We present methods to roughly localize the object and estimate the characteristics needed for grasping. Our work may be seen as an alternative to the grasping procedures that are using a previously built data-base. The thesis is divided in two parts : the rough object localization and the estimation of its characteristics.

Given the coordinates of the clicked point, the object is known to be on the view line which connects both the remote camera optical center and the clicked point. The projection of this view line on the gripper image plane is the epipolar line associated with the clicked point. Epipolar based visual servoing is used to control the embedded camera to scan this line. Image characteristics are extracted from both the remote and the gripper view and then matched to estimate the 3D position of the object. This method holds the advantage of being robust to object motion in the remote frame. At the end of the localization process, the object is included in both fields of view and the estimation of the characteristics is initialized.

The object rough shape estimation is treated with a monocular mobile camera. The object shape is approximated by a quadric whose parameters are estimated from the object projection on a set of images. The object is segmented using an active contour method that is initialized using the output of the localization process. The better the viewpoints, the more accurate the characteristics estimation. Finally, a active vision method is developed to automatically select the viewpoints that improve reconstruction. The best views are chosen in order to maximize the new information.

Keywords : Rehabilitation robotics, Visual servoing, 3D Modelisation, Active Vision