



Contrôle redondant de la position d'un robot par capteurs externes. Applications en milieux médical et industriel

Agnés Poyet

► To cite this version:

Agnés Poyet. Contrôle redondant de la position d'un robot par capteurs externes. Applications en milieux médical et industriel. Autre [cs.OH]. Institut National Polytechnique de Grenoble - INPG, 1996. Français. NNT: . tel-00005012

HAL Id: tel-00005012

<https://theses.hal.science/tel-00005012>

Submitted on 23 Feb 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée par

Agnès Poyet

pour obtenir le grade de Docteur
de l'Institut National Polytechnique de Grenoble
(Arrêté ministériel du 30 Mars 1992)

Spécialité : Informatique

Contrôle redondant de la position d'un robot par capteurs externes

Applications en milieux médical et industriel

Date de soutenance : 23 Janvier 1996

Composition du jury

Président :	M. Radu Horaud
Rapporteurs :	M. Raja Chatila
	M. Olivier Faugeras
Examineurs :	M. Philippe Cinquin
	Mme Marie-Agnès Garnero
	M. David Hawkes
	Mme Jocelyne Troccaz

Thèse préparée au sein du laboratoire TIMC - IMAG
sous la direction de Jocelyne Troccaz

Merci à ...

Merci à Madame Jocelyne Troccaz, alias Super Joce, directrice de thèse cinq étoiles, pour avoir toujours su doser à la perfection enthousiasme et pondération, encadrement et liberté d'action.

Merci aux rapporteurs, Messieurs Raja Chatila et Olivier Faugeras pour tout le temps qu'ils ont passé à la lecture attentive de ce manuscrit et pour l'intérêt qu'ils lui ont accordé.

Merci à Monsieur Philippe Cinquin d'avoir lancé ce projet et de m'avoir accueillie au sein de son équipe.

Merci à Madame Marie-Agnès Garnero d'avoir accepté de faire partie du jury, mettant ainsi un point d'orgue à la collaboration TIMC-EDF.

Merci à Monsieur David Hawkes d'avoir participé à ce jury en dépit des difficultés linguistiques.

Merci à Monsieur Radu Horaud d'avoir bien voulu présider cette soutenance et d'avoir témoigné de nombreuses marques d'intérêt.

Merci à la Direction des Etudes et Recherche de l'EDF de m'avoir financée pendant ces trois années de thèse. Merci en particulier à Monsieur Michel Carré, l'instigateur de la collaboration TIMC-EDF, à Monsieur Jean-Claude Bordas pour avoir repris le flambeau, à Madame Marie-Agnès Garnero et à Monsieur Philippe Cers pour leurs discussions techniques fructueuses en toute sympathie et à Monsieur Lopez pour son coup de main lors des expérimentations sur le site de Châtou.

...

Et parce qu'une thèse, ce n'est pas seulement un travail acharné parachevé par une soutenance, c'est de tout coeur que je remercie mes fidèles compagnons de tablée (voir figure 0.1) et de manière plus générale tous les membres de l'équipe GMCAO, avec une pensée particulière pour notre maman à tous : Maribel.

Merci à

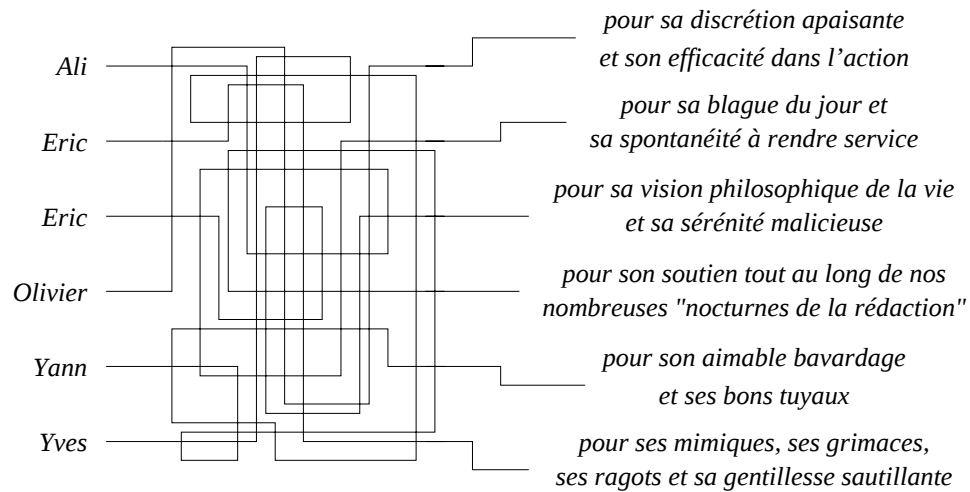


FIG. 0.1 - Merci aux fidèles compagnons de tablée.

Merci à Rachel pour ses encouragements mailistiques et son dynamisme sympathique.

Merci à David et Claude pour m'avoir généreusement prêté leur PC pour mes scéances de rédaction parisiennes. Qu'ils sachent cependant que leur tentative de corruption consistant à installer insidieusement DOOM a été vaine : je n'ai pas craqué!...

Merci à mes parents de m'avoir toujours laissée libre de choisir mes études.

Enfin un grand merci à Bertrand pour son soutien et son attention de tous les jours. Merci à lui d'avoir accepté de jouer le rôle du candide et d'avoir relu ce manuscrit avec minutie et acharnement.

Résumé

Les problèmes de sécurité constituent le principal obstacle à l'utilisation de robots dans des applications sensibles. Nous répertorions ces problèmes et les solutions partielles qui y sont apportées dans divers domaines d'application de la robotique. Nous détaillons en particulier deux applications qui nous intéressent : le guidage d'un geste médico-chirurgical et l'inspection d'un bol de générateur de vapeur dans une centrale nucléaire.

Notre objectif consiste à vérifier de manière globale la bonne exécution de la tâche planifiée. Nous cherchons donc à contrôler la position du robot au moyen de capteurs externes. Aussi étudions-nous les divers types de capteurs de position envisageables avant de proposer une solution qui consiste à apposer des marqueurs passifs sur le robot et à surveiller leur position au moyen de caméras vidéo.

Cette manière de procéder implique le calibrage préalable de chacun des composants du système. Nous étudions en détail les méthodes de calibrage intrinsèque de caméras et adoptons une méthode multi-plans que nous étendons au calibrage de caméras à focale variable. Les caméras étant calibrées intrinsèquement, nous proposons une solution pour les recalibrer par rapport à l'environnement spécifique de chacune de nos deux applications. La dernière phase de calibrage consiste à repérer la position d'un marqueur par rapport au segment du robot qui le porte. Ce problème est tout-à-fait similaire à la détermination de la position d'une caméra portée par un robot. Nous étudions particulièrement deux solutions classiques dont nous améliorons la précision en minimisant une erreur globale sur un ensemble important de données. Toutes ces étapes étant réalisées, la position des marqueurs peut être prédite, puis vérifiée grâce à un traitement d'images approprié.

Des expérimentations complètes ont été menées en vue de montrer la faisabilité du projet et d'évaluer la précision que l'on peut espérer.

mots-clefs : vision, robotique, sécurité, contrôle de position, calibrage de caméras, caméras à focale variable, calibrage main-œil, détection subpixel

Abstract

Safety issues are of prime importance in using robots in some sensitive applications. We classify the possible incidents and the partial solutions that exist in various robotic fields. We particularly detail two applications of interest: the guiding of a medical intervention and the inspection of a steam boiler in a power plant.

We aim at checking, in a global way, the correct accomplishment of the planned task. Thus we attempt to control the robot position by means of external sensors. That is why we study different kinds of position sensors that would be possible and then we suggest a solution that consists in drawing some passive markers on the robot and then in controlling their position by means of video cameras.

To proceed so, it is necessary to first calibrate the different components of the system. The intrinsic calibration of the cameras is studied in detail and we choose a method that uses two calibration planes. We extend this method to cameras with a variable focal length. Once the cameras are calibrated, their position in the environment has to be determined. We suggest one specific solution for both of our applications. The last calibration step consists in determining the position of each marker with respect to the robot link it belongs to. This is a similar problem to the “hand-eye calibration”. We particularly study two classic methods that we improve by minimizing a global error on a sizeable data set. By combining all these steps, we can predict the markers position, and then check it thanks to an appropriate image processing.

Experiments have been conducted in order to prove the feasibility of the project and to estimate the possible accuracy.

key-words : vision, robotics, safety, position control, camera calibration, variable focal length, hand-eye calibration, subpixel detection

Table des matières

Introduction	5
1 Description des problèmes de sécurité en robotique dans ses divers domaines d'application	9
1.1 La robotique industrielle	10
1.1.1 Présentation	10
1.1.2 Robotique et sécurité dans l'industrie	10
1.1.3 Un exemple de besoin : la surveillance de la plaque tubulaire d'un bol de générateur de vapeur dans une centrale nucléaire .	16
1.2 La robotique d'aide aux handicapés	17
1.2.1 Présentation	18
1.2.2 Prise en compte de la sécurité	21
1.2.3 Conclusion	23
1.3 Les Gestes Médico-Chirurgicaux Assistés par Ordinateur (GMCAO) .	23
1.3.1 Présentation	23
1.3.2 Robotique et sécurité dans les applications de GMCAO	28
1.3.3 Un exemple de besoin : la mise en place d'une prothèse de la hanche sans ciment	32
1.4 Conclusion	33
2 Le contrôle d'exécution par capteurs de position externes	35
2.1 Les divers types de capteurs envisageables	35
2.1.1 Les capteurs de type "digitaliseur"	36
2.1.2 Les capteurs permettant de détecter un point précis	41
2.1.3 Le cas particulier des caméras vidéo	45
2.2 Comparaison des deux solutions envisageables	47
2.2.1 Récapitulatif des deux solutions	47
2.2.2 Comparaison des deux méthodes	50
2.3 Notre approche : le contrôle redondant par suivi de marqueurs dans des images vidéo	52
2.3.1 Présentation de la méthode retenue	52
2.3.2 Présentation du rapport : quels sont les problèmes à résoudre?	52

3	Calibrage intrinsèque des caméras	57
3.1	Etude bibliographique	57
3.1.1	Modèle à sténopé	58
3.1.2	Intégration des distorsions dans le modèle à sténopé	62
3.1.3	Méthodes multi-plans	67
3.1.4	Calibrage à focale variable	70
3.1.5	Conclusion - Méthodes retenues	73
3.2	Mise en oeuvre et résultats	73
3.2.1	Présentation du matériel	73
3.2.2	Répétabilité de l'acquisition d'images	74
3.2.3	Précision du calibrage	77
3.3	Conclusion	87
4	Recalage des caméras par rapport à l'environnement	89
4.1	Méthode proposée pour l'application nucléaire	89
4.1.1	Rappel des besoins et conditions de travail	89
4.1.2	Etude bibliographique	90
4.1.3	Méthode proposée	94
4.1.4	Résultats	95
4.2	Méthode proposée pour l'application médicale	96
4.2.1	Rappel des besoins et conditions de travail	96
4.2.2	Méthode proposée	97
4.2.3	Résultats	99
4.3	Conclusion	100
5	Le calibrage des marqueurs	101
5.1	Objectif: résoudre $A.X = X.B$	102
5.1.1	Description du problème	102
5.1.2	Détermination de A	102
5.1.3	Détermination de B	102
5.1.4	Résolution de $A.X = X.B$	104
5.2	Précision de la résolution	108
5.2.1	Précision de A	108
5.2.2	Précision de B	112
5.2.3	Précision de la solution obtenue par les deux méthodes étudiées	113
5.3	Stratégie globale	120
5.3.1	Fonction d'erreur	120
5.3.2	Minimisation de la fonction d'erreur	121
5.3.3	Minimisation globale sur un ensemble de matrices	122
5.3.4	Algorithme	122
5.4	Résultats	123
5.4.1	Données simulées bruitées	123
5.4.2	Données réelles	124
5.5	Remarques et conclusions	125

6	Détection et suivi des marqueurs	127
6.1	Prédiction de la position d'un marqueur	127
6.2	Détection d'un marqueur	129
6.2.1	Influence de la précision de la détection sur la précision du positionnement 3D	129
6.2.2	Etude bibliographique des algorithmes de détection subpixel .	130
6.2.3	Précision	144
6.2.4	Conclusions	145
6.3	Suivi d'un marqueur dans une séquence d'images	147
6.3.1	Suivi effectué dans des conditions idéales	148
6.3.2	Prise en compte des occultations	148
6.3.3	Suivi sans connaissance a priori	150
6.3.4	Conclusion	151
7	Expérimentations	153
7.1	Evaluation de la partie VISION dans le cadre de l'application nucléaire	153
7.1.1	Dispositif expérimental et matériel utilisé	153
7.1.2	Etapes impliquées	154
7.1.3	Résultats	156
7.2	Evaluation du système complet dans le cadre de l'application médicale	157
7.2.1	Dispositif expérimental et matériel utilisé	157
7.2.2	Etapes impliquées	158
7.2.3	Résultats	161
7.3	Conclusion	165
8	Limites et extensions possibles	167
8.1	Spécificités propres à chaque application	167
8.1.1	Recalage par rapport à un repère absolu pertinent	167
8.1.2	Choix concernant les caméras	169
8.1.3	Choix concernant les marqueurs	173
8.2	Ce qu'il reste à faire: la gestion de haut niveau	178
8.2.1	Architecture	178
8.2.2	Rôle du contrôleur central	178
8.2.3	Rôle des processeurs décentralisés	179
8.3	Autres extensions possibles	180
8.3.1	Suppression de l'Optotrak	181
8.3.2	Diagnostic de l'origine de la panne	182
8.3.3	Commande du robot grâce à la vision	183
	Conclusion	191
A	Description du localisateur Optotrak	195
B	Différence entre le milieu de la projection et la projection du milieu	197

Introduction

Un robot est une machine capable de manipuler des objets ou d'exécuter des opérations selon un programme. La notion de programmabilité permet à un utilisateur d'adapter le robot à des variations mineures de l'environnement et de lui faire réaliser des tâches variées.

Les robots ont été initialement conçus, d'une part pour affranchir l'homme des tâches répétitives, d'autre part pour améliorer la productivité. Puis ils ont été substitués à l'homme pour la réalisation de tâches dans des environnements hostiles : hautes pressions (eaux profondes), basses pressions (espace), hautes températures (chaufferies), basses températures (cryogénie) et fortes radiations (au contact de matériels nucléaires ou dans l'espace). La nature de leur travail a donc évolué. Il ne s'agit plus pour eux de simplement répéter une tâche apprise ; ils doivent également être capables de réaliser une tâche "réfléchie". Pour cela, ils ont besoin d'une autonomie croissante. Aussi sont-ils dotés de capteurs de plus en plus nombreux et sophistiqués qui augmentent leurs capacités cognitives. D'autre part, il est nécessaire qu'ils présentent une bonne précision et non plus seulement une bonne répétabilité.

Le champ d'action des robots s'est ensuite étendu à des domaines pour lesquels ils n'avaient pas été pensés initialement. C'est le cas du domaine médical et en particulier de la chirurgie. Le plus souvent, le chirurgien définit avant l'intervention, une stratégie opératoire à partir des images médicales. Alors le rôle du robot consiste généralement à matérialiser cette stratégie pour aider le chirurgien à la réaliser.

Du fait de la nature particulièrement délicate des tissus nerveux, la neurochirurgie a exigé très tôt un système pour positionner précisément l'outil par rapport à la cible à atteindre. C'est ce qui a conduit à l'utilisation d'un cadre stéréotaxique fixé sur le crâne du patient. Ce cadre constitue un intermédiaire dans lequel on peut repérer, d'une part la cible au moyen de l'imagerie, d'autre part l'outil grâce à une conception mécanique précise. L'ajustement de la position de l'outil s'est d'abord fait manuellement, grâce à des verniers, avant d'être "robotisé". Les systèmes robotisés, allant du cadre stéréotaxique motorisé au manipulateur de robotique industrielle, ont ainsi été introduits dans les salles d'opération au cours des années quatre-vingts.

La flexibilité des robots manipulateurs industriels les rend potentiellement utilisables pour une large variété d'interventions, mais ils doivent bien sûr subir des modifications afin de limiter les risques engendrés par leur action à proximité, et même au contact, d'êtres humains. Dans le contexte industriel, les risques de ce

Introduction

type sont quasiment inexistants puisque toute présence humaine est généralement bannie du champ d'action du robot durant son fonctionnement. L'adaptation des robots industriels au domaine médical est donc délicate. D'ailleurs peu de systèmes robotisés industriels ont été utilisés dans des applications cliniques. On peut cependant citer le pionnier en la matière : le système RobodocTM destiné à usiner le fémur pour la mise en place d'une prothèse de la hanche.

Récemment, des robots spécialement conçus à des fins chirurgicales ont vu le jour. Certains sont dédiés à une intervention donnée. Leur architecture engendre des mouvements naturels similaires aux mouvements nécessaires pour réaliser la tâche. D'autres facilitent la transmission de la stratégie planifiée au chirurgien, grâce par exemple à des systèmes de retour d'efforts. Ces systèmes robotiques sont intrinsèquement plus sûrs que les robots manipulateurs de type industriel. Malgré tout, il semble très important d'adjoindre à tous ces systèmes robotisés, des mécanismes logiciels et matériels de détection de pannes et de réaction à celles-ci.

Dans cette thèse, nous nous intéressons à l'utilisation de capteurs externes afin de vérifier que le robot exécute précisément la trajectoire planifiée à l'avance. Ceci a pour but d'augmenter la sécurité lors de l'utilisation de robots, tant dans le domaine industriel que dans le domaine chirurgical.

Le **premier chapitre** sera consacré aux problèmes de **sécurité**, ainsi qu'aux solutions partielles qui y sont apportées, dans trois domaines d'application de la robotique : l'industrie, l'aide aux personnes handicapées et les gestes médico-chirurgicaux assistés par ordinateur.

Nous distinguerons deux types de comportement du robot en cas de problème : soit une déviation importante, et la plupart du temps brusque, soit une dérive faible, par rapport à la trajectoire attendue.

Le dysfonctionnement d'un composant électronique peut être à l'origine du premier type de comportement. Dans le domaine industriel, les conséquences graves vis-à-vis de l'homme sont supprimées en imposant que les espaces de travail de l'homme et du robot soient en permanence disjoints. Dans les deux autres domaines étudiés, pour lesquels des robots industriels ont été utilisés sans être véritablement adaptés, des architectures spécialisées semblent être la solution la plus prometteuse.

Le deuxième type de comportement peut être dû à un mauvais calibrage du robot, à des jeux mécaniques ... Ces erreurs sont plus délicates à appréhender car elles ne se manifestent pas de manière flagrante. Néanmoins leurs conséquences pouvant être dramatiques, il est primordial de les détecter. Nous illustrerons ce besoin par deux exemples d'application : l'examen de la plaque tubulaire d'un bol de générateur de vapeur d'une centrale nucléaire et la mise en place d'une prothèse de la hanche.

Nous nous proposons alors d'effectuer un contrôle de la position d'un robot au moyen d'un capteur externe. Nous étudions **divers types de capteurs** au **chapitre 2** et nous proposons un système répondant aux exigences générales d'un contrôle redondant des mouvements d'un robot, et aux exigences particulières de nos deux

exemples d'application.

Ce système consiste à surveiller le robot avec un ensemble de caméras vidéo. Le robot est équipé de dessins géométriques, que nous dénommons marqueurs, et dont la position par rapport au segment qui le porte peut être connue par une procédure de calibrage. Les caméras sont calibrées intrinsèquement et sont recalées par rapport à l'environnement. Il est alors possible, à partir d'un modèle du robot et des valeurs de ses codeurs, de prévoir la position théorique des marqueurs dans l'espace, puis de quantifier la cohérence entre cette position théorique et les informations fournies par la vision.

La fin du chapitre présente l'articulation des principaux modules du système qui seront détaillés aux chapitres 3, 4, 5 et 6.

Le **chapitre 3** concerne le **calibrage intrinsèque des caméras**. Après une étude bibliographique, nous décrivons les tests de précision que nous avons effectués avec la méthode retenue : méthode des deux plans avec une modélisation cubique. Nous étendons cette méthode au cas du calibrage d'un module de comescope à focale variable et accomplissons le même type de tests.

Le **recalage des caméras** par rapport à l'environnement est traité au **chapitre 4**. Les deux exemples d'application étudiés dans cette thèse présentent des environnements très différents. Aussi nous proposons une solution particulière pour chacun d'eux.

Pour l'application nucléaire, nous adaptons et optimisons une méthode de localisation de caméra par rapport à un repère dans lequel sont définis un ensemble de points de référence.

Pour l'application médicale, nous adoptons une approche pragmatique qui consiste à effectuer le calibrage intrinsèque des caméras directement dans le repère absolu, à savoir dans ce cas, le repère associé à la base du robot. Nous comparons les résultats de ce calibrage avec ceux du chapitre précédent.

Nous explicitons le **calibrage des marqueurs** au **chapitre 5**. Celui-ci revient à résoudre un système d'équations matricielles de la forme $A_i X = X B_i$. Nous étudions deux méthodes classiques de résolution : la méthode de Tsai [TL88], reprise par Horaud [Hor], et la méthode de Shiu et Ahmad [SA87]. Nous évaluons ces deux approches sur des données simulées ce qui nous permet de définir les conditions que doivent remplir les données pour ne pas mettre les méthodes de résolution en défaut. Nous décrivons la stratégie que nous avons adoptée et qui consiste à minimiser une fonction d'erreur sur un ensemble important de données. Nous évaluons cette stratégie sur des données simulées bruitées et nous déterminons de manière indirecte la précision obtenue sur des données réelles.

Les étapes décrites précédemment sont des étapes préliminaires à la phase de surveillance du robot. La vérification proprement dite est effectuée grâce à la **détection** et au **suivi des marqueurs** dans les images. La première partie du **chapitre 6** explicite comment prédire une position approximative des marqueurs dans les

Introduction

images. La précision de la détection étant primordiale, nous étudions et classifions les méthodes de détection subpixel de points dans les images. Nous choisissons alors la forme de nos marqueurs et optons pour un algorithme qui présente un bon compromis entre la rapidité d'exécution et la précision de la détection.

Enfin, nous étudions le suivi d'un marqueur et en particulier les informations que peut nous fournir un tel suivi pour prévoir une occultation du marqueur.

Le **chapitre 7** décrit les deux **expérimentations** que nous avons menées. La première concerne uniquement la partie “vision” du système et a été mise en œuvre dans des conditions proches de celles impliquées par une application du type de l'application nucléaire. La seconde intègre toutes les phases du système dans le cadre de l'application médicale. Elle permet ainsi de valider notre approche.

Le **chapitre 8** est consacré à une **analyse du projet** dans son ensemble.

Nous examinons d'abord les spécificités de chaque application et les contraintes qu'elles impliquent, en particulier dans le choix des caméras et des marqueurs. Nous détaillons ces choix dans le cas de nos deux exemples d'application.

Nous fournissons ensuite un aperçu de ce que pourrait être la gestion de haut niveau des informations fournies par chacune des caméras d'un système de contrôle redondant, puis nous décrivons d'autres extensions possibles du projet.

Enfin la conclusion présente une brève synthèse du travail accompli et des résultats obtenus.

Chapitre 1

Description des problèmes de sécurité en robotique dans ses divers domaines d'application

Nous allons étudier trois domaines d'intervention des robots, les problèmes de sécurité qui en découlent et les mesures mises en place pour les prévenir :

- **La robotique industrielle** : Elle concerne non seulement les tâches répétitives, telles que de l'assemblage ou de la manutention, mais aussi des tâches plus complexes que l'homme ne peut pas réaliser lui-même car elles se déroulent en milieux hostiles tels que dans une centrale nucléaire, dans l'espace, sous la mer ...

Ainsi la robotique industrielle recouvre un domaine très vaste. Une analyse générale des difficultés est donc très délicate. Cependant, le point crucial concernant la sécurité vis-à-vis des humains a été résolu en minimisant les interactions entre l'homme et le robot.

En revanche, persiste le problème de sécurité vis-à-vis du matériel, tout particulièrement pour des opérations requérant une grande précision.

Les robots ont été créés à des fins industrielles. Cependant, ils ont été détournés de leurs fonctions premières vers d'autres usages. C'est le cas des deux domaines que nous décrivons ensuite. Les problèmes de sécurité proviennent en grande partie du fait que les robots utilisés n'ont pas été spécifiquement conçus pour s'acquitter des tâches qui leur sont assignées.

- **La robotique d'aide aux handicapés** : Dans ce domaine, la précision n'est pas de toute première importance. En revanche, le robot ne s'avère utile que s'il entre en contact avec son utilisateur, par exemple pour lui apporter un objet, l'aider à manger ... La sécurité vis-vis de l'homme ne peut donc pas être résolue en supprimant les interactions.
- **La robotique des Gestes Médico-Chirurgicaux Assistés par Ordinateur** : Il s'agit d'aider à la réalisation de gestes médico-chirurgicaux par

des systèmes de guidage robotisés. Aussi ce domaine est-il particulièrement sensible aux problèmes de sécurité puisqu'il combine les impératifs des deux autres domaines : précision dans la réalisation de la tâche et forte interaction avec l'homme (patient et personnel médical).

1.1 La robotique industrielle

1.1.1 Présentation

C'est le tout premier domaine d'application de la robotique, du point de vue de la chronologie, et du point de vue de l'étendue et de la diversité des utilisations potentielles. Les robots industriels sont principalement dédiés à des tâches répétitives qui consistent simplement à reproduire une trajectoire apprise auparavant. La répétabilité constitue alors le premier critère d'évaluation de la qualité du robot employé. Les robots peuvent également être utilisés pour réaliser une tâche en milieu hostile. Ces interventions sont plus ou moins complexes et nécessitent souvent une programmation hors ligne du robot. La précision peut alors être de toute première importance.

1.1.2 Robotique et sécurité dans l'industrie

Nous allons nous intéresser aux problèmes et aux solutions liés à l'utilisation de robots en milieu industriel. Nous nous restreindrons aux aspects de détection et de prévention des risques vis-à-vis des hommes et du matériel. Nous n'aborderons pas dans son intégralité l'aspect "sûreté de fonctionnement" dont le but consiste à maintenir la production, en qualité et en quantité, même en cas de défaillances. Pour une étude globale de la sûreté de fonctionnement, on pourra se reporter à [NS89].

Les accidents auxquels nous allons nous intéresser peuvent être classés en trois catégories selon le degré de responsabilité du système robotisé¹ :

- Les accidents dûs à une mauvaise utilisation du robot : l'accident n'est pas directement imputable au système robotisé, mais sa présence l'a rendu possible.
- Les accidents dûs à un comportement anormal du robot (déviation large et/ou brutale de sa trajectoire).
- Les accidents dûs à une piètre précision du robot.

1. On appelle système robotisé, l'ensemble constitué par le robot lui-même (carcasse, moteurs, capteurs associés ...), le système de commande (matériel et logiciel) et les moyens de communication entre les deux.

1.1.2.1 Problèmes et solutions liés à une mauvaise utilisation du robot

Dans ce cas, les accidents ne sont pas directement imputables au système robotisé, mais à la manière de l'utiliser.

Les législations européenne et française ont établi des normes réglementant les sites de production comprenant des robots manipulateurs industriels. Ces différentes normes concernent la sécurité en général [Afn93a], la prévention des accidents d'origine mécanique [Afn93c], la prévention des accidents d'origine électrique [Afn93b] ... Les consignes de sécurité doivent être adaptées au type de robot utilisé, à ses possibilités de commandes et à ses conditions d'utilisation. Aussi nous n'allons énoncer que des règles générales concernant le mode d'utilisation d'un système robotisé.

La technologie de la commande de robots étant essentiellement de type électronique, elle s'avère sensible aux conditions d'utilisation. Les circuits imprimés et les unités de mémoire sont susceptibles d'être altérés par la poussière (particules magnétiques), l'humidité, les hautes températures, la pollution, les chocs et vibrations, les bruits, les interférences électriques et les variations de tension. Aussi, la première catégorie de précautions à prendre concerne l'environnement matériel du robot.

Diverses protections doivent être envisagées pour prévenir les conséquences de ces perturbations : filtrage, isolation, support anti-vibrations, tests de mémoire ...

D'autres précautions doivent être prévues dès la conception du robot, au cas où les protections vis-à-vis de l'environnement ne rempliraient pas totalement leurs fonctions. Ainsi, en cas d'incident électrique, le robot doit être freiné automatiquement et garder sa stabilité, c'est-à-dire se maintenir en position même s'il n'est plus alimenté.

La deuxième catégorie de mauvaises conditions d'exploitation de la robotique concerne le comportement des utilisateurs. On peut distinguer deux modes d'utilisation d'un robot :

- une utilisation de routine,
- une utilisation hors production.

L'utilisation de routine est la phase durant laquelle le robot se contente d'exécuter une tâche apprise ou programmée auparavant. L'homme n'a donc nul besoin d'être en contact avec le robot. Les mesures de sécurité élémentaires consistent à définir l'espace atteignable par le robot et à en interdire le fonctionnement tant qu'un utilisateur est présent dans cet espace.

Pour cela, on peut mettre en place autour du robot une cage dont l'ouverture coupe l'alimentation du robot. On peut également avoir recours à divers détecteurs de présence reliés à l'armoire de commande du robot. Ces capteurs peuvent par exemple détecter des variations de pression ou de température, des mouvements, des changements de propriétés électriques du milieu. Ils se divisent en deux catégories :

les détecteurs matériels (contacteur, fil sensible, contour de sécurité, grilles, filets, tapis à contact) et les dispositifs sensibles de présence (détecteur de proximité, rayon lumineux, rideau lumineux, champ capacitif, détecteur à micro-ondes, à ultrasons, détecteur infrarouge, caméra).

Le champ des principes physiques permettant de détecter sans contact une présence (humain ou objet) est très large, mais l'obtention d'un signal fiable en toute occasion est souvent difficile. Il peut alors être intéressant de coupler différents capteurs pour augmenter la fiabilité de la détection d'autant plus que leur sensibilité aux perturbations est susceptible d'engendrer de fausses alertes nuisibles au bon fonctionnement et à la production. D'autre part, [RDW93] propose un système de surveillance décentralisé basé sur les informations fournies par des caméras CCD (Charged Couple Device). La distribution des traitements parmi les noeuds d'un système complètement décentralisé rend celui-ci très robuste et lui permet d'assurer sa survie même en cas de dysfonctionnement de l'un des capteurs.

On dénote très peu d'accidents en utilisation de routine car les précautions précédemment décrites sont faciles à mettre en oeuvre. Les accidents surviennent principalement lors des phases de maintenance ou de réparation. Ceci s'explique par une méconnaissance des risques du fait qu'il s'agit d'interventions ponctuelles d'une part, et par un contact inévitable entre l'homme et la machine d'autre part.

Les phases d'apprentissage sont également propices aux accidents du fait que l'opérateur doit se maintenir à proximité du robot pour surveiller la trajectoire à enregistrer. Ceci se fait généralement au moyen d'un clavier portatif qui permet de valider l'acquisition de points de passage de l'organe terminal (outil).

La vitesse de déplacement du robot est limitée durant l'apprentissage (0.25 m/s en Europe; 0.30 m/s aux Etats-Unis et au Japon). Notons que l'on peut également limiter la vitesse maximale de chacun des organes du robot, en phase d'apprentissage aussi bien qu'en phase de routine, afin de diminuer l'énergie cinétique du robot et donc d'atténuer les conséquences en cas de problème. La limitation de la vitesse est pénalisante pour la réalisation de l'apprentissage car il est plus facile pour l'utilisateur d'appréhender la trajectoire si elle est effectuée à vitesse normale. En revanche elle laisse davantage de temps à l'opérateur pour réagir en cas d'incident. D'ailleurs les normes actuelles imposent au clavier portatif d'être conçu ergonomiquement de sorte que le bouton d'arrêt d'urgence soit aisément accessible. Les effets de ce bouton devraient être immédiats. Hélas, en général, l'arrêt n'est pas instantané à cause de l'inertie du robot.

D'autre part, le clavier dispose d'une touche d'homme-mort qui permet de valider l'activation des déplacements du robot. La sécurité impliquée par cette touche repose sur le fait qu'elle exige une action positive de la part de l'utilisateur, pour déplacer le robot et non pas pour l'arrêter. En effet, le robot ne se déplace que lorsque cette touche est maintenue enfoncée par l'utilisateur. Ceci permet non seulement d'accélérer la réaction de l'opérateur en cas de contre-temps, mais également de diminuer les erreurs d'inattention.

Enfin l'aspect ergonomique du système de commande est de toute première importance : Les couleurs, les formes, les emplacements et le texte utilisés doivent être étudiés avec attention pour éviter les ambiguïtés et donc les erreurs.

Dans le cas où la programmation du robot ne se fait pas par apprentissage, mais par programmation hors ligne, la présence d'un homme n'est pas requise dans le champ d'action du système robotisé. Cependant, une mauvaise programmation du robot peut induire des problèmes de sécurité vis-à-vis du matériel environnant. Ce type d'erreur peut être prévenu par des techniques non spécifiques de la robotique telles que l'évaluation de la fiabilité d'un logiciel [Gau90] ou les preuves de programme (voir [PTP87] pour une application de ces techniques à la gestion des incertitudes dans le cas d'un programme de planification de tâches robotisées). D'autre part, la mise au point d'un tel programme peut être assistée par un simulateur qui permet de visualiser les effets du programme avant de l'exécuter sur le robot lui-même.

1.1.2.2 Problèmes et solutions liés à un comportement anormal du robot

Certains incidents peuvent engendrer un comportement anormal du robot. Ce comportement anormal se traduit par des mouvements incontrôlés et à forte inertie à cause des capacités mécaniques du robot. Les systèmes de sécurité ont alors pour objectifs de détecter l'incident et de stopper le robot avant qu'il n'entre en collision avec l'environnement. Une déviation brutale de la trajectoire du robot peut provenir d'une mauvaise programmation du robot en amont, comme nous l'avons vu précédemment, ou d'un problème électronique.

Les problèmes électroniques ont pour origine le dysfonctionnement d'un des composants. Certains de ces problèmes peuvent être détectés par des vérifications simples sur les données transmises ("check-sum", bit de parité ...) ou par la duplication du système d'acquisition des données (codeur et communication). Les deux ensembles de valeurs acquises sont théoriquement identiques. La constatation d'une différence entre eux constitue un signal d'alarme qui doit déclencher l'arrêt du robot.

De manière plus générale, c'est la duplication globale du système qu'il faut envisager afin de comparer les valeurs pertinentes à chaque étape de l'exécution de la tâche, l'objectif final étant de disposer d'un système tolérant les pannes. En toute rigueur, cela signifierait que tous les capteurs et tous les moyens de communication sont dupliqués; toutes les procédures de traitement de ces données sont écrites dans des langages différents par des équipes différentes (pour éviter des fautes de conceptions identiques) et tournent sur des machines électriquement indépendantes ayant des systèmes d'exploitation différents. Enfin, en cas de détection de problème, chaque pendant de l'application doit être en mesure d'arrêter le robot indépendamment l'un de l'autre.

Cette solution n'est cependant envisageable que pour des applications critiques en matière de sécurité, applications qui justifient alors le surcoût engendré (dans le domaine spatial ou le domaine nucléaire, par exemple). Davies [Dav93] a préco-

nisé, sans la réaliser, cette solution pour des applications de chirurgie assistée par ordinateur (domaine que nous étudierons ultérieurement). D'une manière générale les applications pour lesquelles il serait raisonnable de prendre de telles mesures sont celles dans lesquelles une erreur pourrait engendrer des conséquences catastrophiques.

Ainsi la duplication totale d'un système n'est quasiment jamais réalisée en pratique. En revanche une duplication partielle est possible. Cependant, on peut difficilement ajouter de nouveaux éléments sur des robots déjà opérationnels. Ces précautions doivent donc être envisagées dès la conception de nouveaux robots, si l'on ne veut pas que le surcoût d'une telle sécurité ne soit prohibitif.

Un autre type de solution pour la détection d'un comportement anormal consiste à utiliser des capteurs externes. Cette solution permet de détecter des incidents sans avoir besoin d'identifier exactement leur origine. En cela, elle est plus générale puisqu'elle détecte des incidents qui n'ont pas forcément été prévus, en observant leurs conséquences, plutôt qu'en traquant leurs causes : un capteur d'accélération pourra détecter un problème interne à l'origine d'un emballement du robot; des capteurs de proximité pourront détecter la présence anormale d'un objet à proximité du robot et donc prévenir les collisions par une réduction de la vitesse pouvant aller jusqu'à l'arrêt complet. Notons que le problème d'évitement d'obstacles est courant en robotique et que de nombreux algorithmes apportent une solution lorsque le champ d'action du robot est connu. Des représentations sophistiquées permettent également de traiter ces problèmes en temps réel dans un environnement mouvant [SH92].

Une fois qu'un problème a été détecté, il s'agit d'arrêter le robot le plus rapidement et le plus sûrement possible. Le moyen le plus radical d'arrêter le robot consiste à couper son alimentation. Cependant, cela implique d'assurer mécaniquement que le robot reste stable et que le système de préhension, qui permet de tenir un objet ou un outil, reste activé en cas d'absence de puissance.

Cet arrêt peut être déclenché automatiquement par des fusibles de vitesse activés par le fluide hydraulique dans le cas de robots hydrauliques et par des limiteurs de courant dans le cas de robots électriques.

Cependant l'arrêt du robot est d'autant plus lent que son inertie est importante. On a donc tout intérêt à la minimiser. Pour cela, on peut jouer sur l'emplacement des éléments lourds (moteurs, engrenages) par rapport aux articulations, sur la forme géométrique du bras et sur ses matériaux constitutifs.

L'énergie cinétique d'un robot emballé peut également être maîtrisée grâce à des butées mécaniques et des absorbeurs de chocs, ou par l'utilisation d'un freinage dynamique.

La sécurité intrinsèque peut aussi être améliorée mécaniquement dès la conception par l'élimination des points de coincement, l'absence d'arrêtes vives et de saillies, l'encastrement des tuyaux et des cables et par la mise en place de protecteurs sur

les mécanismes d'entraînement.

Enfin, l'algorithme du système de commande peut intervenir dans la sécurité intrinsèque par la définition de l'erreur de poursuite tolérable². La force de collision entre le robot et un obstacle étant proportionnelle à cette erreur, une réduction de sa valeur dans l'algorithme de commande renforce la sécurité.

Les moyens de détection d'erreur précédemment décrits sont adaptés à des problèmes de type "rupture" (matériel cassé, mauvais traitement des données), mais ne répondent pas à des problèmes dont les symptômes sont moins flagrants, et se traduisent par exemple par une dérive progressive de la trajectoire du robot.

1.1.2.3 Problèmes et solutions liés à une précision médiocre du robot

Un tel phénomène peut être dû soit à des jeux trop importants dans les articulations, suite à une usure des matériaux, soit à un changeur de coordonnées³ du robot inadapté à l'utilisation actuelle. Cela se produit lorsque le robot est utilisé dans des conditions différentes de celles dans lesquelles on l'a calibré⁴. Le phénomène se manifestera par exemple avec un robot calibré dans des conditions normales de pression et de température, et utilisé dans l'espace (basse pression) ou en cryogénie (basse température). Ce phénomène apparaît également dans des situations plus courantes : on utilise un robot afin de transporter une lourde charge, alors qu'il a été calibré à vide. Les déformations transversales des éléments du robot, engendrées par la charge, modifient sensiblement son comportement cinématique, ce qui se traduit par une piètre précision dans l'exécution de la tâche.

Il est évident que ce genre d'erreur ne peut pas être détecté par des capteurs internes du robot. Il faut avoir recours à des moyens de mesure externes pour contrôler l'exécution de la tâche. Différents types de capteurs envisageables seront étudiés en détails dans le chapitre suivant.

1.1.2.4 Conclusion

Ainsi, le problème de sécurité vis-à-vis de l'homme a été résolu en limitant au strict minimum ses interactions avec le robot. En ce qui concerne les problèmes, tels que les risques de collision avec l'environnement, ils peuvent être détectés grâce à des capteurs attachés au robot lui-même (capteurs d'accélération, de proximité) et

2. L'erreur de poursuite tolérable est l'erreur limite au-delà de laquelle on estime que l'asservissement des moteurs ne peut plus être maintenu. Ce seuil permet de déceler certains dysfonctionnements et de commander l'arrêt du robot.

3. Le changeur de coordonnées du robot est la transformation qui permet de passer des valeurs articulaires du robot (valeurs des codeurs de chaque articulation) aux coordonnées cartésiennes (position et orientation par rapport à un repère absolu) de l'outil fixé sur son dernier degré de liberté.

4. Calibrer un robot consiste à estimer précisément les paramètres de son changeur de coordonnées.

leurs conséquences peuvent être atténuées par des précautions dès la conception du robot (coupure de courant automatique, minimisation de l'inertie ...)

En revanche, la détection des erreurs d'imprécision lors de l'exécution de la tâche implique un contrôle redondant externe. Nous allons détailler ces besoins pour une application donnée dans le paragraphe suivant.

1.1.3 Un exemple de besoin : la surveillance de la plaque tubulaire d'un bol de générateur de vapeur dans une centrale nucléaire

D'un point de vue très synthétique, une centrale nucléaire est constituée de deux circuits (voir figure 1.1) :

- le circuit primaire d'eau radioactive,
- le circuit secondaire qui alimente les turbines en vapeur.

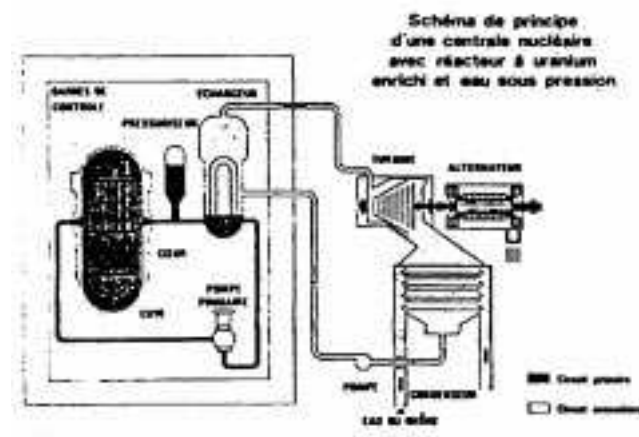


FIG. 1.1 - *Schéma de principe d'une centrale nucléaire*

L'échange de chaleur entre les deux circuits se fait au niveau du générateur de vapeur, ou échangeur : l'eau radioactive arrive dans la première partie du bol de générateur de vapeur (GV), en forme de quart de sphère ; elle circule ensuite dans plus de 5000 tubes ce qui permet de chauffer l'eau du circuit secondaire et de la transformer en vapeur ; enfin, elle est évacuée dans la deuxième partie du bol de GV. La vapeur du circuit secondaire active des turbines reliées à un alternateur, transformant ainsi l'énergie nucléaire en énergie électrique.

L'étanchéité de l'interface entre les deux circuits est primordiale pour des raisons évidentes de sécurité vis-à-vis de la radioactivité. Aussi est-elle vérifiée à chaque phase de maintenance de la centrale. Le contrôle de la plaque tubulaire, du côté primaire, se fait actuellement par l'homme avec toutes les contraintes que cela implique : l'opérateur doit prendre toutes les précautions nécessaires au travail en atmosphère

radioactive ; le contrôle doit être fait sans erreur sur des milliers de tubes. Aussi, l'EDF a envisagé d'automatiser cette tâche au moyen d'une caméra portée par un robot, ce qui permettrait d'augmenter la sûreté de la vérification et de diminuer notablement la durée de présence des opérateurs humains en atmosphère radioactive, leur intervention consistant simplement à fixer le robot au plafond du bol de GV et à installer quelques autres outils dans le bol.

Le premier problème à résoudre dans cette approche consiste à s'assurer qu'en aucun cas, le robot n'entre en collision avec l'environnement, ce qui implique entre autre, de vérifier la cohérence des informations de position renvoyées par les codeurs afin de prévenir les conséquences d'incidents tels qu'un codeur cassé ou une communication défectueuse.

L'autre point délicat de cette approche est la précision atteignable par le robot. Les tubes ont un diamètre d'environ 2 cm et la distance entre les centres de 2 tubes contigus est de 2.74 cm ou 3.25 cm selon le type de générateur. La précision du robot doit donc être meilleure que $2.74/2 = 1.37\text{ cm}$ afin d'assurer qu'on ne fait pas d'erreur sur le numéro du tube examiné. Une précision de l'ordre du centimètre semble une exigence raisonnable. Une fois l'approche grossière effectuée, les manœuvres plus fines telles que l'insertion de la caméra à l'intérieur du tube pourront être réalisées grâce à des méthodes de commandes locales telles que la commande référencée par la vision. Notons qu'à l'heure actuelle, une précision de l'ordre du centimètre n'est pas assurée dans 100% des cas avec le matériel employé par l'EDF, à savoir un robot Shilling. Il est bien entendu qu'une précision encore meilleure serait appréciée et faciliterait le travail d'observation de la plaque tubulaire.

L'évaluation de la précision par un système de contrôle redondant externe va se heurter à plusieurs contraintes spécifiques à cette application :

- Le système de surveillance doit être placé à l'intérieur même de l'espace à surveiller, à savoir le bol de GV. Ceci implique d'une part que le système de surveillance couvre un large champ, puisqu'il ne peut pas avoir de recul, et d'autre part qu'il soit peu volumineux afin de ne pas encombrer le bol de GV.
- Le robot doit être équipé le moins possible, d'une part pour ne pas augmenter le coût, d'autre part pour ne pas gêner les manipulations du robot lors de son installation dans le bol. Cette installation par des opérateurs humains doit en effet être rapide puisqu'elle s'effectue en milieu radioactif.

1.2 La robotique d'aide aux handicapés

Nous avons vu que dans le cadre industriel, la sécurité vis-à-vis de l'homme a été résolue en supprimant, dans la mesure du possible, les interactions entre l'homme et le système robotisé. Cependant cette règle n'est plus applicable dans des domaines où justement l'objectif du robot consiste à opérer près d'un être humain pour l'aider. C'est le cas des deux autres domaines d'intervention de la robotique que nous allons

étudier : la robotique pour handicapés et les Gestes Médico-Chirurgicaux Assistés par Ordinateur.

1.2.1 Présentation

La robotique peut intervenir de deux façons distinctes pour aider les personnes handicapées.

La plus spectaculaire consiste à suppléer, voire à remplacer totalement la fonction atteinte. Pour cela, on a recours à des orthèses, c'est-à-dire des systèmes robotiques placés à l'extérieur des membres endommagés, ou à des prothèses, qui elles se substituent aux membres amputés.

L'un des premiers projets d'orthèse active est dû à l'initiative du Professeur Rabbishong de l'INSERM de Montpellier [MR84]. Il a pour objectif originel de permettre à des paraplégiques de marcher avec l'aide d'une orthèse motorisée autonome, puis à terme, de restituer la fonction de marche.

Quant aux prothèses, la difficulté majeure dans la réalisation de leur fonction motrice réside dans la communication avec le handicapé. La technologie actuelle permettrait de concevoir des prothèses très sophistiquées, mais les commandes envisageables restent limitées à cause du nombre réduit de signaux que l'homme est capable d'émettre avec une partie localisée du corps.

Un grand espoir réside dans le développement de prothèses nerveuses implantables, basées sur la capacité de régénérescence du système nerveux périphérique. Cette nouvelle génération de prothèses permettrait d'établir une continuité électrique bidirectionnelle entre le système nerveux et une instrumentation externe.

Une communication dans le sens "système nerveux $\rightarrow$ instrumentation" résoudrait les délicats problèmes de contrôle de prothèse par la personne handicapée. Il s'agit d'exploiter le système nerveux valide pour commander une prothèse ou une orthèse. Une communication dans le sens "instrumentation $\rightarrow$ système nerveux" permettrait de rendre leur motricité à des gens dont la moëlle épinière a été endommagée (voir [BHIV94] pour une description de quelques expériences concrétisées). Il s'agit alors de s'affranchir de la partie du système nerveux endommagée et d'envoyer des commandes aux muscles encore valides. C'est l'objet du projet EUREKA CALIES (Computer-Aided Locomotion by Implanted Electro-Stimulation) qui a l'ambition de stimuler de manière coordonnée une cinquantaine de muscles afin de restituer une marche quasi naturelle [HB93].

La majorité de ces projets très prometteurs n'est pas opérationnelle. Une approche plus modeste, mais plus réaliste à l'heure actuelle, est représentée par le développement de robots assistants.

Les robots assistants ont pour objectifs d'aider les handicapés dans leur vie quotidienne. Ils représentent une assistance plus économique que la présence continue d'une aide-soignante, et surtout ils donnent à leur utilisateur une autonomie appréciable. Ils sont capables de saisir un objet (verre, tasse ...) et de l'apporter à proximité de l'utilisateur ; Handy 1 donne à manger [Top93]. Ils peuvent gratter,

relever une mèche et aider à faire la toilette. Master [DLB91] et DeVAR [Pa90] donnent de plus à leurs utilisateurs la possibilité d'effectuer un travail de bureautique (introduire une disquette dans un lecteur, prendre des feuilles dans une imprimante, tourner les pages d'un livre ...).

L'étendue des tâches que ces systèmes sont capables de réaliser dépend en grande partie de leur architecture.

1.2.1.1 Architectures des robots assistants

On peut classer les systèmes existants selon quatre types d'architecture :

– **Station de travail statique** (Tou [CVC93], Isac [Kaw94])

La station de travail, le robot et les périphériques sont regroupés en un seul endroit et n'en bougent pas. Tout est connu à propos de l'environnement, en particulier les positions des objets à manipuler. Son espace de travail étant entièrement structuré, le robot peut opérer de manière répétitive, ce qui ne requiert aucun effort de contrôle de la part de l'utilisateur.

D'un point de vue technique, c'est la solution la plus simple à réaliser. Cependant elle oblige l'utilisateur à vivre constamment au même endroit de la maison.

– **Robot monté sur un rail** (Master [DLB91], DeVAR [Pa90])

Il s'agit simplement d'une extension du premier type de système. Etant monté sur un rail, le robot peut atteindre un espace de travail plus grand.

Ces deux types d'architecture sont intéressants pour des personnes sévèrement handicapées, puisqu'elles leur permettent d'accomplir des tâches sans nécessiter un contrôle précis de leur part. Cependant, tous ces systèmes peuvent également fonctionner selon un mode interactif qui laisse davantage d'autonomie à l'utilisateur.

– **Robot monté sur un fauteuil roulant** (Manus [Ra91b], Inventaid [HPH92])

Cette architecture autorise plus de souplesse puisque le robot se déplace avec son utilisateur.

Cependant la place restreinte sur le fauteuil roulant limite la taille du robot et de son contrôleur. L'architecture adaptée de Manus lui permet malgré tout d'atteindre un espace de travail important. Il peut entre autre toucher le sol grâce à sa base cylindrique pivotante et télescopique qui permet un déplacement vertical de l'épaule.

D'autre part, l'environnement n'étant plus structuré, le robot doit être guidé interactivement pour accomplir la tâche demandée. Les difficultés rencontrées lors de la préhension d'objets, peuvent être surmontées avec l'aide d'un système de vision artificielle.

Notons enfin que le fauteuil roulant lui-même, peut être “intelligent”. Certains fauteuils sont en effet équipés de capteurs leur permettant d’éviter les collisions. On peut citer le fauteuil du CEA qui adapte sa vitesse à la distance de l’obstacle détecté le plus proche [GDLM93]. D’autres, tels que le fauteuil de Manus, sont capables de monter et descendre des escaliers.

- **Robot mobile** (MoVAR [VdLML86], URMAD [DGMDN93] et MOVAID [Gug94])

C’est la solution la plus complexe, mais aussi la plus générale. Elle nécessite la coordination de sous-systèmes dédiés à la vision, au mouvement, à la manipulation fine ...

MoVAR (Mobile Vocational Assistant Robot) a été spécifié dès 1988, mais il n’est pas encore utilisable avec une autonomie complète.

URMAD, acronyme des mots italiens signifiant Unité Robotique Mobile pour l’Assistance aux Handicapés, est une plate-forme de recherche intégrant l’état de l’art des technologies robotiques. Ce système à huit degrés de liberté montés sur une base mobile, sera autonome dans un environnement partiellement connu et reconnaîtra les commandes de l’utilisateur exprimées dans un langage synthétique de haut niveau. Cependant le véhicule de URMAD est trop grand et trop gros. MOVAID a pour objectif de s’affranchir de ces handicaps en répartissant dans l’environnement de travail, les fonctions, le poids, l’alimentation, les capteurs et les capacités de calcul.

1.2.1.2 Moyens de contrôle

Pour tirer le meilleur parti des apports de la robotique, il faut des moyens de communication et de contrôle adaptés aux capacités motrices de l’utilisateur.

Une personne sévèrement handicapée a besoin de tâches de haut niveau pré-programmées et doit pouvoir les commander par un minimum d’interaction avec le système. Lorsque l’utilisateur est capable de parler, un système simple de reconnaissance vocale, limité à quelques mots, constitue une interface pratique [Kaw94, CVC93, Pa90]. Les personnes handicapées peuvent également utiliser n’importe quel moyen de contrôle moteur en leur possession (main, bouche, tête, souffle ...). Ils activent alors un interrupteur unique qui sélectionne des commandes à partir d’un système de balayage dont on peut régler la vitesse en fonction de l’utilisateur. Ce genre de système est connu sous le terme anglais de “scanning switch system” [DLB91, Bir93].

[Bir93] propose également un clavier agrandi pour les utilisateurs capables de déplacements, mais disposant d’un faible contrôle en précision.

Enfin, tous ces systèmes mettent à la disposition des utilisateurs une manette permettant soit de sélectionner des tâches pré-programmées, soit de commander directement les déplacements du robot.

Pouvoir commander directement le robot procure davantage d’autonomie, mais

peut s'avérer fastidieux lorsqu'il s'agit de tâches complexes que l'utilisateur est amené à réitérer. C'est pourquoi la plupart des systèmes permettent la création et la mémorisation de nouvelles tâches grâce à un langage approprié de haut niveau. A titre d'exemple, on peut se référer à [DMJ93] qui décrit CURL (Cambridge University Robot Language), un langage de haut niveau destiné à la fois à des programmeurs et à des utilisateurs non informaticiens. Voir également les langages utilisés par DeVAR [Lei91] et Master [DLB91].

Du fait de la proximité de l'homme, le contrôle du robot ne doit autoriser aucune collision dangereuse, tout en laissant en permanence la maîtrise du système à l'utilisateur. Les moyens de contrôle doivent intégrer, dès leur conception, les mesures de sécurité à prendre en compte pour espérer une utilisation sans risque des robots assistants. Le paragraphe suivant présente les différentes manières d'aborder le problème de la sécurité.

1.2.2 Prise en compte de la sécurité

La plupart des systèmes ont intégré les mesures élémentaires d'arrêt d'urgence dans les procédures de contrôle du robot. Il s'agit la plupart du temps d'une interruption déclenchée par une touche du clavier ou bien d'un ou plusieurs boutons d'arrêt immédiat atteignables de manière sûre par l'utilisateur. Celui-ci doit donc surveiller continuellement le robot pendant ses déplacements; il doit disposer de capacités cognitives suffisantes pour se rendre compte d'un problème et avoir un minimum de motricité pour intervenir rapidement.

Cependant ce genre de système requiert de la part de l'utilisateur une action positive pour localiser le bouton puis pour l'activer, ce qui implique un délai non négligeable. Un moyen de réduire ce délai consiste à exiger une action positive pour autoriser le robot à se déplacer plutôt que pour l'arrêter (interrupteur d'homme mort). Par exemple, l'utilisateur appuie sur un interrupteur tant que le mouvement du robot lui convient. Pour arrêter l'action, il lui suffit de relâcher l'interrupteur. Cette fonction est facile à implémenter pour des opérateurs capables de contrôler leurs efforts, mais délicate pour des opérateurs dont les capacités sont réduites, typiquement lorsque le contrôle se fait via un "scanning switch system".

Plutôt que réduire le délai d'intervention, on peut laisser davantage de temps à l'utilisateur pour réagir en diminuant la vitesse de déplacement du robot. Cependant, cette vitesse doit rester suffisamment grande pour apporter une aide réelle et un confort d'utilisation. Une vitesse trop faible aurait pour conséquences de démotiver les utilisateurs potentiels de tels systèmes et de relâcher l'attention des utilisateurs persévérants. La sécurité ne doit pas être un frein à l'ergonomie des systèmes.

Les opérateurs n'étant pas toujours capables de détecter un problème ou d'intervenir rapidement, il est bon de suppléer leurs lacunes en analysant les données de divers types de capteurs. Les capteurs de proximité et de force peuvent déceler une collision [Pa90], alors que les capteurs de vision sont adaptés à la détection de mouvements soudains de l'utilisateur qui peuvent requérir l'arrêt du robot [Kaw94]. Une précaution supplémentaire consiste à doubler les codeurs et les moyens de com-

munication permettant ainsi de s'assurer de la validité des informations de position fournies par le robot [Ra91b].

Un moyen d'éviter les collisions du robot avec l'utilisateur consisterait à séparer leurs enveloppes de travail respectives. Ceci n'est malheureusement pas envisageable puisque c'est contraire aux objectifs des robots d'aide aux handicapés qui exigent des interactions avec l'utilisateur. Pour les stations de travail statiques ou pour les robots montés sur un rail, on peut limiter la zone d'interaction au strict minimum [CDLL94], mais cela restreint les apports du robot. En effet, le seul type d'aide possible avec cette architecture consisterait à ce que le robot prenne un objet dans son champ d'action, l'amène et le dépose dans la zone commune puis se retire de la zone commune pour laisser l'utilisateur attraper l'objet déposé. Ce type de précaution ne semble donc pas adapté au domaine de la robotique d'aide aux handicapés.

On peut également diminuer les risques en limitant volontairement les capacités du robot. [Bir93] préconise de restreindre l'amplitude de chaque articulation en effectuant une vérification continue par programme et en désactivant le robot en cas de dépassement. Cependant la sécurité est renforcée si les limites sont mécaniques. Dans le même ordre d'idées, Manus [Ra91b] et Inventaid [HPH92] ont des couples de forces limités.

Enfin, la technologie utilisée pour mouvoir le robot est de première importance. En effet, la technologie pneumatique (Inventaid) permet une certaine souplesse (ou "compliance" dans le jargon de la robotique) que n'autorise pas la puissance électrique (Manus). Elle est donc mieux adaptée pour travailler à proximité de l'homme.

Voici deux exemples de robots assistants dont la conception présente un degré de sécurité élevé du fait que leur structure supporte des déformations en cas de collision :

- ISAC [Kaw94] a un squelette rigide activé par plusieurs paires de muscles. Ces muscles sont en fait des tubes en caoutchouc renforcés de fibres. Leur longueur dépend de la pression de l'air à l'intérieur des tubes. Le contrôle est similaire à celui des muscles humains.
- Tou [CVC93] est intrinsèquement sûr grâce à sa structure molle. Il est constitué de six modules cylindriques en caoutchouc mousse. Le mouvement du bras est produit par la déformation adéquate de chaque module par des fils métalliques internes. L'angle de chaque déformation est limité à 30 degrés pour prévenir les irréversibilités et les hystérésis excessifs. Des pliures plus importantes peuvent apparaître lorsque des forces extérieures agissent sur le robot, ce qui est un indice de sécurité.

Dans ces deux cas, le guidage n'est pas très précis puisqu'il s'effectue par l'intermédiaire de structures susceptibles d'être déformées. Il nécessite donc une implication importante de l'utilisateur pour contrôler le robot en boucle fermée. Cependant celui-ci est aidé par des systèmes auxiliaires : un système de vision artificielle (Isac,

Tou) permet d'affiner le contrôle du robot lors de manoeuvres requérant de la précision (préhension d'un objet, approche de nourriture à proximité de la bouche ...) alors qu'un pendule dans le poignet du robot assure le maintien de l'horizontalité (Tou).

1.2.3 Conclusion

Contrairement au domaine de la robotique industrielle, le domaine des robots assistants n'a pas de standard de sécurité à respecter. Il faut dire que les interactions entre l'homme et le robot rendent difficile la définition de lois relatives à la sécurité.

La sécurité est bien sûr une priorité dans la conception d'un système robotique. Cependant, il ne faudrait pas que des règles strictes limitent de manière trop sensible ces nouveaux moyens offerts aux personnes handicapées.

Les robots à structure molle, équipés de capteurs auxiliaires, semblent constituer un bon compromis.

1.3 Les Gestes Médico-Chirurgicaux Assistés par Ordinateur (GMCAO)

1.3.1 Présentation

Le but ultime des GMCAO est de réaliser "un partenariat entre un homme (le chirurgien) et des machines (ordinateurs et robots) qui vise à exploiter les aptitudes des deux pour réaliser une tâche mieux que ne pourrait le faire chacun d'eux séparément"⁵. Les points cruciaux pour parvenir à cette collaboration fructueuse sont de deux types :

- **Une exploitation optimale des informations** fournies par les divers moyens d'imagerie. Il s'agit, d'une part, d'exploiter quantitativement les informations de position, d'autre part, d'améliorer la qualité des données grâce à la fusion d'images d'origines diverses. Muni de ces images multi-modales, le chirurgien est alors en mesure d'établir un diagnostic mieux fondé, puis de définir une stratégie opératoire précise. Dans des cas complexes, il peut être aidé par le système qui va chercher à optimiser un critère donné (trajectoire d'un outil pour rester le plus loin possible de structures sensibles telles que les vaisseaux sanguins ou les nerfs ; en radiothérapie, schéma d'irradiation qui concentre son action sur la tumeur en épargnant les structures saines adjacentes ...).
- **Une réalisation précise de la stratégie** précédemment décrite. Cette étape implique, d'une part, la transposition dans le champ opératoire de la stratégie définie précédemment sur les images pré-opératoires, et d'autre part, sa réalisation à proprement parlé.

5. Traduction libre d'un extrait de [TCK⁺91]

La phase de transposition revient à déterminer la transformation géométrique entre le repère dans lequel a été définie la stratégie (repère associé aux systèmes d'imagerie pré-opératoires) et le repère dans lequel doit être exécutée la dite stratégie (repère associé au patient par l'intermédiaire d'un moyen d'imagerie léger ou d'un localisateur placé près de la table d'opération). Le calcul de cette transformation repose sur les mêmes méthodes que celles employées pour la création des images multi-modales, à savoir des techniques de mise en correspondance.

Le résultat de cette phase de transposition est exploitée pour mettre en place un système de guidage qui va aider le chirurgien dans son geste opératoire. Un tel guidage lui permet en effet d'accroître sa précision et de réaliser des interventions complexes, par exemple sur des organes difficiles à atteindre ou peu visibles.

Les GMCAO améliorent donc potentiellement la qualité des interventions en les rendant plus faciles, plus précises, moins invasives et quelquefois plus rapides. De nombreuses applications chirurgicales (correction de scoliose [Sau94], ligamentoplastie [OLJ⁺93], neurochirurgie [LCDB91] ...) peuvent entrer dans le cadre des GMCAO dont nous allons détailler uniquement le point technique qui nous concerne, à savoir les systèmes de guidage.

Ces systèmes sont destinés à assister le chirurgien lors de l'exécution de l'intervention. On peut les classer selon trois catégories en fonction de l'autonomie qu'ils laissent au chirurgien : les systèmes passifs, semi-actifs ou actifs.

1.3.1.1 Les systèmes passifs

Ces systèmes laissent une autonomie complète au chirurgien. Des capteurs per-opératoires lui permettent de contrôler son action par rapport aux données pré-opératoires. On distingue deux types de systèmes passifs :

- **Les navigateurs** : Dans ce cas, aucune stratégie n'a été définie auparavant. Le chirurgien voit une représentation symbolique de l'extrémité de son outil sur des images pré-opératoires réelles ou sur des coupes reformatées en temps réel en fonction de la position de l'outil [BCC95]. Le navigateur lui permet donc d'augmenter ses capacités de localisation par rapport aux images à l'origine du diagnostic.
- **Les “systèmes à but”** : Ces systèmes sont nécessaires lorsque le choix de la stratégie est trop difficile à effectuer en direct. Leur apport principal consiste à rendre possible une comparaison entre la stratégie prévue et sa réalisation véritable. L'interface de tels systèmes se décompose en deux parties :
 - La première partie est utilisée avant l'opération pour définir un planning opératoire. Par exemple, l'interface peut permettre de choisir une coupe

reformatée selon la direction désirée, puis d'y définir une trajectoire rectiligne [LCDB91];

- La deuxième partie fournit des informations pour reproduire la stratégie précédemment définie. D'une part, elle présente les mêmes fonctionnalités qu'un navigateur; d'autre part, elle fournit des données pertinentes par rapport aux exigences du planning opératoire. Par exemple, elle peut simplement indiquer la distance entre la position courante de l'extrémité de l'outil et la cible à atteindre. Cependant une telle indication est difficilement interprétable. Aussi on a recourt à des interfaces plus explicites : [LTS⁺96] propose de représenter par une croix les deux points intéressants (extrémité de l'outil et cible à atteindre). L'objectif est réalisé lorsque les deux croix sont superposées. Dans cet exemple, le chirurgien essaie de respecter l'alignement des deux croix, mais il est libre d'intégrer d'autres contraintes. Le système est purement passif.

Dans les deux cas, un localisateur tri-dimensionnel est bien sûr indispensable afin de repérer l'outil. Il peut être optique⁶, tel que le système canadien Optotrak de Northern Digital, ou bien mécanique⁷.

Le grand avantage de ces systèmes est qu'ils laissent une autonomie complète au chirurgien. La contre-partie est que la précision atteinte grâce à ces systèmes est limitée à la précision manuelle du chirurgien dans le suivi des indications.

Ainsi, on ne peut pas vraiment dire qu'un système passif "guide" le chirurgien : il lui fournit simplement un retour d'information en direct. L'outil est donc localisé, mais il n'en est pas moins délicat à mettre et à maintenir en place. Il est en effet très difficile d'aligner un outil selon six degrés de liberté (les trois translations et les trois rotations de l'espace) avec seulement un retour d'information en position. Il est encore plus délicat de maintenir un outil selon une direction donnée tout en exerçant simultanément une force (par exemple lorsqu'on perce un trou).

Pour remédier à ce problème d'ergonomie, on s'est orienté vers des systèmes semi-actifs.

1.3.1.2 Les systèmes semi-actifs

Ces systèmes contraignent physiquement l'action du chirurgien, mais ne la réalisent pas eux-mêmes, contrairement aux systèmes purement actifs. Ainsi [Ra91a] a ajouté des freins sur chacun des axes d'un manipulateur⁸. Ces freins sont actionnés à 10 ou 20 % de leurs capacités lorsque l'utilisateur est proche de la pose⁹ finale

6. Localisateurs optiques : une série de marqueurs actifs est observée par plusieurs caméras ; leur détection dans les différentes images et une simple triangulation permettent de calculer leur position 3D ; les marqueurs sont en fait des diodes lumineuses synchronisées avec l'acquisition des images par les caméras.

7. Localisateurs mécaniques : Ce sont de simples bras articulés sans moteur. On peut calculer la position de l'outil fixé à l'extrémité du bras à partir des valeurs renvoyées par les codeurs.

8. Un manipulateur est un bras articulé non motorisé.

9. Pose : position et orientation.

et ils sont complètement bloqués lorsque celle-ci est atteinte. Ce système simplifie énormément le positionnement d'un objet par le chirurgien et permet la réalisation de trajectoires rectilignes. En effet, l'objet positionné peut être un guide mécanique dans lequel le chirurgien n'a plus qu'à insérer son outil pour effectuer l'action (typiquement, exercer une force sur l'outil pour l'introduire dans le corps selon une trajectoire rectiligne). Le recours à un robot actif pour positionner le guide à distance, constitue une amélioration appréciable pour le chirurgien [Lav89].

[DHN⁺92] a conçu un système mécanique qui permet d'effectuer des mouvements coniques. Ce système est utilisé dans des opérations de résection de la prostate (l'objectif de l'opération est la création d'un vide à l'intérieur de la prostate). Le système possède trois degrés de liberté : l'outil de coupe est placé sur un arc circulaire, lui-même monté sur un cercle libre de tourner de 360 degrés. L'arc conjugué avec le cercle permet à l'outil coupant d'effectuer un mouvement conique en pivotant à peu près sur son extrémité. D'autre part, ce système peut être translaté selon l'axe du cône pour engendrer une suppression progressive de la matière. Une série d'emplacements successifs sur l'arc, le cercle et l'axe de translation permet d'effectuer en séquence des coupes contrôlées.

Le système a été robotisé en vue d'améliorer la rapidité d'exécution. Il peut fonctionner soit en combinant les mouvements des moteurs pour produire la trajectoire requise, soit en effectuant une séquence de mouvements ne requérant que l'activation d'un seul axe à la fois, en vue d'accroître la sécurité.

La résection de la prostate ne nécessite pas une trop grande précision (2 ou 3 mm sont suffisants). Ces systèmes, qui limitent mécaniquement la zone atteignable, mais n'imposent pas une trajectoire rigoureuse, sont donc bien adaptés à ce type d'opérations. Il existe d'ailleurs de nombreuses approches très similaires par exemple pour la laparoscopie [TFLT92, ULWW94] ou pour la chirurgie oculaire [PSMP91] qui impliquent elles aussi des mouvements coniques.

Une approche complètement différente est celle adoptée par l'Université d'Aachen grâce au recours à des gabarits (ou "templates") individuels [RSR94] : les gabarits sont des objets définis et manufacturés avant l'opération par un système de CAO. L'une des faces de cet objet est le négatif de la surface de l'os sur lequel va être effectuée l'intervention. Le rôle de la face opposée (face extérieure) sera décrit ultérieurement. D'autre part, le gabarit remplit une fonction de guide par l'intermédiaire d'une perforation qui le traverse de part en part et dont l'emplacement et la forme sont déduits de la définition de la stratégie en phase pré-opératoire.

Durant l'opération, l'objet est encastré sur l'os. L'étape de mise en correspondance s'effectue donc de manière tout-à-fait naturelle. L'usinage pratiqué dans le gabarit indique la trajectoire à suivre et contraint mécaniquement les déplacements de l'outil lors de l'intervention chirurgicale.

Cet usinage peut être un simple trou cylindrique, auquel cas la trajectoire réalisée sera rectiligne ; un disque placé autour de l'outil permet de limiter mécaniquement la profondeur de l'insertion en butant contre la surface extérieure du gabarit. Un usinage plus élaboré permet de définir une trajectoire plane. La perpendicularité entre

l'outil et la surface extérieure du gabarit est là encore assurée par un disque placé autour de l'outil. On peut imaginer la réalisation de trajectoires plus sophistiquées grâce à la combinaison d'une hauteur variable de la surface extérieure du gabarit et de différentes inclinaisons de perforation. Les trajectoires envisageables demeurent malgré tout relativement simples par rapport aux trajectoires possibles grâce à un robot motorisé.

L'élimination des aspects informatiques de la phase per-opératoire constitue bien sûr l'intérêt majeur de cette approche. On supprime l'étape de mise en correspondance et le guidage se fait naturellement permettant ainsi de s'affranchir d'un bon nombre de problèmes d'ergonomie et de sécurité. Toutes les difficultés techniques (conception du gabarit à partir des images pré-opératoires et de la stratégie) sont réalisées avant l'opération. Elles n'ont donc pas de critères de temps réel à respecter.

En revanche, cette méthode présente l'inconvénient d'être relativement invasive puisqu'elle impose au chirurgien de pratiquer une large ouverture pour pouvoir poser le gabarit sur la structure anatomique de référence qui doit être rigide. Les gabarits ne sont donc pas utilisables sur des structures molles. Notons cependant qu'un projet de "gabarit à distance" est en cours d'étude [Wag95]. Ce projet permettrait de positionner le gabarit dans l'espace au moyen d'un robot. Cette approche combine l'idée d'un simple guide positionné par un robot et celle d'un guide plus sophistiqué constitué par un gabarit.

Ainsi, les systèmes de guidage semi-actifs résolvent le problème de positionnement précis. Malgré cela, ils ne sont pas entièrement satisfaisants : les guides classiques ne permettent que des trajectoires simples ; les gabarits sont relativement invasifs puisqu'ils imposent de dégager complètement la structure de référence pour les mettre en place ; leur utilisation est limitée aux structures osseuses et pour le moment, ils ne permettent eux aussi de réaliser que des trajectoires simples.

On a déjà vu que les systèmes actifs pouvaient faciliter le positionnement d'un guide. Ils sont d'autant plus adaptés à la réalisation de trajectoires complexes.

1.3.1.3 Les systèmes actifs

Dans ce cas, des sous-tâches de la stratégie sont exécutées avec l'aide d'un système robotique autonome, supervisées par le chirurgien et contrôlées par des capteurs redondants. Ces systèmes sont dédiés à la réalisation d'une tâche complexe. Il existe bien sûr des différences entre une cellule robotisée dans le domaine industriel et une "cellule robotisée chirurgicale". [KMZ⁺93] décrit les modifications apportées à un robot industriel pour le rendre utilisable dans le milieu chirurgical. Cette étude porte sur le système RobodocTM développé par l'équipe de Taylor [TPC⁺92]. Robodoc est à notre connaissance le premier robot actif sans contrainte qui ait été utilisé avec succès sur plusieurs patients. Il a pour but d'augmenter les performances des chirurgiens dans les procédures d'usinage précis d'un os en chirurgie orthopédique, avec comme application initiale, le remplacement total de la hanche sans ciment. Les résultats ont été analysés et ont montré une amélioration effective de la précision

par rapport à une opération manuelle sans aide.

Le robot Minerva [GFE⁺95] effectue des opérations neurochirurgicales. Il a été spécialement conçu (matériau, configuration géométrique ...) pour travailler à l'intérieur d'un scanner ce qui permet de fournir à tout moment, un retour d'information au chirurgien. L'opération est entièrement réalisée par le robot. Sa précision est de 0.5 mm alors qu'elle est de 2 mm lorsqu'elle est exécutée manuellement.

Un autre robot a été conçu pour effectuer les mêmes opérations mais sous le contrôle d'un système IRM [MKM⁺95]. Les contraintes sont plus fortes à cause de la présence de champs magnétiques importants. Ce système encore très récent présente une précision globale de 3 mm.

Les systèmes actifs s'avèrent donc les plus performants pour réaliser rapidement des tâches complexes avec une bonne précision. Cependant, la prise en charge totale de l'exécution de la tâche par le robot, au détriment du chirurgien, ne répond pas à notre philosophie des GMCAO. Elle prive le chirurgien de la maîtrise totale des opérations et du coup augmente les risques : une action effectuée par le chirurgien et contrôlée par le système est plus sûre qu'une action effectuée par le système (le robot en l'occurrence) et passivement contrôlée par le chirurgien. Dans le deuxième cas, il n'est en effet pas possible d'assurer que le chirurgien reste concentré en permanence sur ce qu'est supposée faire une machine très sûre, de telle sorte que le temps de réaction à un incident non détecté par le système peut être très lent, avec éventuellement des effets tragiques. Au contraire, dans le premier cas, le chirurgien est en charge de l'action ; donc ses capacités à détecter un incident inattendu et à réagir rapidement sont exploitées au maximum.

C'est pourquoi dans le paragraphe suivant, où nous discutons de la sécurité dans les applications de GMCAO, nous allons dissocier les risques engendrés spécifiquement par les systèmes actifs et les problèmes plus généraux des systèmes de guidage, à savoir essentiellement la précision.

1.3.2 Robotique et sécurité dans les applications de GMCAO

On a vu dans la présentation des Gestes Médico-Chirurgicaux Assistés par Ordinateur que ce domaine combine les difficultés de la robotique industrielle (précision indispensable dans la réalisation de certaines tâches) et celles de la robotique d'aide aux handicapés (sécurité indispensable lors de l'utilisation de systèmes actifs à proximité d'humains). Ainsi les deux paragraphes suivants vont reprendre les remarques déjà faites pour les deux domaines précédents et vont développer les points spécifiques aux GMCAO.

1.3.2.1 Problèmes et solutions liés à l'utilisation de systèmes actifs

Du fait qu'il est actif, le robot a les capacités d'agir de lui-même ce qui peut causer des dommages s'il suit des indications erronées ou est soumis à des fluctuations

d'alimentation.

En ce qui concerne les problèmes engendrés par de mauvaises conditions d'utilisation du robot (surtensions, vibrations ...), on peut se reporter au paragraphe 1.1.2.1.

En cas de problèmes constatés par l'utilisateur, les boutons d'arrêt d'urgence et les interruptions par clavier sont un bon moyen d'intervention. Tout comme pour les deux autres domaines, les boutons doivent être bien visibles et facilement accessibles. De plus, le système doit être conçu de telle sorte qu'une interruption soit possible à tout instant dans le déroulement du logiciel.

Cependant le système est encore plus efficace s'il interagit en permanence avec l'utilisateur. Ceci peut être réalisé au moyen de demandes de confirmation avant l'exécution de chaque étape comportant des risques, ou bien au moyen d'un système d'interrupteur d'homme-mort. Ce système est plus facilement utilisable que dans le cas de la robotique d'aide aux handicapés, dans la mesure où l'utilisateur est en pleine possession de ces capacités motrices. Il faudra cependant tenir compte des problèmes de stérilisation lors de la conception de cet interrupteur. Dans le domaine des GMCAO, on peut imaginer d'installer un tel système sur l'effecteur du robot. Ainsi l'opérateur tiendrait l'extrémité du robot tout en l'amenant en position. La capacité de l'utilisateur de sentir les forces et les changements soudains d'accélération ou de vitesse lui permettrait d'avoir un temps de réaction rapide en lâchant l'interrupteur d'homme-mort.

Cependant l'utilisateur n'est pas toujours en mesure de détecter une erreur et donc d'intervenir. C'est pourquoi, il faut mettre en place des procédures de détection d'erreur capables d'envoyer un signal d'arrêt au robot et des procédures d'arrêt capables de stopper le robot sans risque. D'autre part il faut anticiper les conséquences engendrées par les erreurs potentielles afin de les prévenir.

Les moyens de détection d'une erreur seront étudiés dans le paragraphe suivant concernant le contrôle de la bonne réalisation de la tâche.

Que l'arrêt du robot soit commandé par l'utilisateur (appui sur un bouton d'urgence) ou par le contrôleur central du système suite à une détection d'erreur, il doit s'effectuer de manière sûre, à la fois rapide et sans brusquerie. On pourra se référer au paragraphe 1.1.2.2 pour des remarques concernant la minimisation de l'inertie du robot. Notons cependant que le domaine des GMCAO est moins sujet aux fortes inerties que l'industrie, car la vitesse d'exécution et les charges à supporter sont bien moindres.

D'autre part, les procédures d'arrêt doivent permettre de reprendre l'opération là où elle en était ou bien autoriser le retrait du robot et la poursuite manuelle de l'intervention sans le recours à la méthodologie des GMCAO [TKMP90, DNH93].

Une reprise sans danger de l'opération implique entre autre de s'assurer que lorsque le système est réinitialisé, il n'a pas bougé par rapport au patient. Une alternative consiste à s'assurer qu'on ramène le robot à la main vers une position de départ connue pour le réinitialiser. Ces mesures sont bien sûr indispensables en cas de détection d'erreur, soit par l'utilisateur, soit par le système.

Indépendamment de ces procédures d'arrêt d'urgence, l'utilisateur doit avoir la possibilité de faire une pause à tout moment afin de s'assurer de la bonne marche des opérations en cours.

Il faut également anticiper les conséquences fâcheuses même en cas d'incident non prévisible. Un moyen simple de minimiser ces conséquences consiste à limiter mécaniquement les capacités du robot. On peut agir sur la vitesse en utilisant des réducteurs importants [LCDB91] ce qui présente l'intérêt de laisser davantage de temps pour réagir. On peut également éviter que le robot n'impose des forces trop importantes en cas de collision, en limitant les couples de forces. On peut restreindre par logiciel, ou encore mieux mécaniquement, le champ d'action du robot en le contraignant à rester dans une région de sécurité prédéfinie. Une restriction par logiciel consiste à surveiller continuellement les valeurs des codeurs du robot et à stopper le mouvement lorsque ces valeurs sortent de l'intervalle prédéfini. Cependant cette manière d'opérer n'intègre pas les éventuelles erreurs des codeurs ou de la transmission de leurs valeurs. C'est pourquoi il est préférable d'avoir recours à des contraintes purement mécaniques. C'est ce que préconise [DHN⁺92] (voir les systèmes de guidage semi-actifs en 1.3.1.2).

Cependant, la notion de région peut être trop large pour des interventions nécessitant le suivi d'une trajectoire avec une très grande précision, comme c'est le cas pour la neurochirurgie.

Contrairement à la robotique d'aide aux handicapés, la robotique des GMCAO n'a pas pour objectif de suppléer des déficiences motrices, mais plutôt de communiquer au chirurgien des informations de trajectoires calculées informatiquement et de lui fournir un surplus de précision dans leur réalisation. Ainsi la fonction motrice du robot n'est pas utilisée pour elle-même, mais comme moyen de transmission d'information. Il semble donc raisonnable d'essayer de la remplacer par un autre type d'interface, sans risque, ce qui résoudrait tous les problèmes de sécurité liés à l'utilisation d'un robot actif. C'est ce qui est en cours d'étude et de développement dans le projet de Robot à Sécurité Passive [DT95] au laboratoire TIMC. Le Robot à Sécurité Passive est un système mécanique de guidage qui tente de combiner les avantages des systèmes actifs et passifs. Ainsi il allie la complexité des tâches potentiellement réalisables par un système actif aux risques limités impliqués par un système passif. Le principe de base est celui de la roue libre.

Dans le prototype actuel (qui ne comporte que deux degrés de liberté), les vitesses sont imposées dynamiquement grâce à des moteurs. Il faut cependant bien noter que le rôle des moteurs n'est pas d'engendrer le mouvement, mais uniquement

de le limiter. C'est le chirurgien qui, en tenant son outil fixé à l'extrémité du robot, fournit l'énergie nécessaire au déplacement du robot. Ainsi en cas d'incidents électriques, ou de défaillances des moteurs, il n'y a aucun risque de déplacement imprévu du robot, le chirurgien restant toujours maître des opérations. Le bras ne pouvant pas bouger de lui-même, aucun risque supplémentaire n'est engendré par son utilisation. Ainsi la responsabilité de l'action est laissée au chirurgien de manière naturelle.

Ce type de système permet donc de résoudre tous les problèmes de sécurité liés à l'utilisation d'un robot actif tout en offrant la possibilité d'effectuer des tâches complexes. Cependant, les problèmes de contrôle d'exécution persistent. En effet, la commande des moteurs est effectuée en fonction de diverses informations : valeurs fournies par les codeurs du robot, changeur de coordonnées du robot, estimation de la cible à atteindre, de la trajectoire à suivre ou de la région de sécurité à respecter. Une erreur dans l'une de ces informations conduit à une mauvaise réalisation de la tâche.

1.3.2.2 Problèmes et solutions liés à une mauvaise réalisation de la tâche

Le premier problème consiste à détecter que la tâche courante est en train d'être mal réalisée. Divers moyens de détection doivent être mis en oeuvre pour faire face aux diverses sources potentielles d'erreur. Le dysfonctionnement d'une partie matérielle (hardware) du système constitue la première catégorie d'erreurs envisageables. Il peut s'agir d'une défaillance d'un codeur, d'une mauvaise communication des données, de parasites électriques ... Heureusement le coût de fabrication de nombreux composants électroniques a considérablement baissé au cours des dernières années. Ceci rend donc possible la mise en place d'un système hardware tolérant les fautes, grâce à la redondance des capteurs (voir paragraphe 1.1.2.2 et [DNH93]). Ainsi la duplication des codeurs et des moyens de communication fournit un moyen simple de détecter une panne.

Ces mesures permettent de détecter directement des erreurs matérielles. Elles ne peuvent en aucun cas détecter des erreurs moins flagrantes telles que le mauvais calibrage du bras articulé ou des jeux mécaniques trop importants qui engendrent des erreurs de positionnement et d'orientation de l'effecteur du robot, bien que les valeurs des codeurs soient exactes. Ces erreurs peuvent avoir pour conséquence le déplacement de l'outil à un endroit dangereux et l'application de forces excessives en cas de collision avec des organes voisins de la cible. Pour remédier à ces problèmes, on peut avoir recours à des mécanismes qui contraignent l'outil à rester dans une certaine région (voir paragraphe précédent). On peut également utiliser des capteurs externes tels que des capteurs de proximité (détection d'obstacles proches) ou des capteurs de force (détection de résistance imprévue). Ce type de détection n'est pas toujours approprié au domaine des GMCAO puisque l'outil est destiné à travailler dans un environnement contraint (à l'intérieur du corps) et qu'il est amené à exercer des forces pour remplir son rôle (percer un trou par exemple). Dans ce cas, il peut

s'avérer délicat d'estimer si les obstacles détectés et les forces exercées sont normales ou non.

1.3.2.3 Conclusion

La véritable requête des GMCAO est un système de guidage qui permettrait d'indiquer aux chirurgiens la trajectoire à suivre, aussi complexe soit-elle. Les systèmes actifs constituent l'une des possibilités. Leur principal avantage est leur capacité à contrôler et à améliorer la vitesse d'exécution d'une tâche, mais cette valeur ajoutée n'est significative que pour de rares applications.

D'autres solutions, non motorisées, donc intrinsèquement plus sûres, existent pour guider le geste du chirurgien en lui imposant des trajectoires simples (guides mécaniques pour trajectoires rectilignes ou coniques) ou en restreignant les mouvements à des régions de sécurité (contraintes mécaniques). Le Robot à Sécurité Passive ou d'autres systèmes basés sur des contraintes dynamiques (systèmes à freins) sont des solutions prometteuses.

Ces approches résolvent les problèmes qu'engendrerait l'utilisation de systèmes actifs. Cependant, tout comme dans le domaine industriel, il persiste des problèmes dûs à une réalisation imprécise de la tâche. Nous allons détailler ces besoins dans le cadre de la mise en place d'une prothèse de la hanche.

1.3.3 Un exemple de besoin : la mise en place d'une prothèse de la hanche sans ciment

La mise en place d'une prothèse au niveau de la hanche intervient en particulier en cas de fracture du col du fémur. Cette opération nécessite de creuser le fémur afin d'y insérer la prothèse. Sa réussite dépend du bon placement de l'implant relativement à l'os, et de la précision avec laquelle la cavité fémorale peut être préparée pour correspondre à la forme de l'implant.

Dans [TPC⁺92], une interface permet au chirurgien de spécifier sur les images de scanner, l'endroit où il souhaite placer l'implant relativement à l'anatomie du patient. Ce positionnement est ensuite transposé dans le repère d'un robot qui est chargé d'usiner dans le fémur, une cavité de forme et de taille exactes à l'endroit spécifié.

Le premier type de problème possible est dû au fait que le robot est actif. Diverses précautions ont été prises [TKMP90] : le contrôleur du robot impose des limites en position et en vitesse dans les boucles d'asservissement articulaire ; l'alimentation du bras est coupée si le contrôleur ne vérifie pas l'intégrité du système toutes les 16 ms ; un capteur de force est monté sur le poignet du robot, ce qui permet de détecter si celui-ci s'emballe. D'autre part, ce système pourrait être remplacé par un système passif du type du bras passif à contraintes dynamique [DT95].

Cependant il persisterait toujours des problèmes de précision. La précision de la réalisation du geste doit en effet être meilleure que le millimètre. Ceci implique d'une part de s'assurer que l'os à usiner ne bouge pas, ce qui est fait grâce à un tensiomètre, d'autre part de vérifier que l'outil de coupe porté par le robot, ne s'éloigne pas de la position demandée. Pour cela, un localisateur 3D Optotrak (voir annexe A) surveille un ensemble de diodes infrarouges fixées sur l'effecteur du robot. Ce système, malheureusement très onéreux, ne serait pas envisageable pour des opérations de routine.

Il existe donc un réel besoin d'un système de contrôle redondant de la position du robot qui n'implique pas de matériel coûteux. Dans un bloc opératoire, un tel système devra tenir compte de l'encombrement de l'espace par le personnel hospitalier. En revanche, il pourra exploiter le fait que l'outil porté par le robot se déplace lentement dans un volume très restreint, et selon des orientations peu différentes.

1.4 Conclusion

Nous avons vu que l'utilisation de la robotique est susceptible d'engendrer deux types de dysfonctionnements :

- Les **déviations brusques** en cas d'anomalies électriques (surtensions, coupure de courant ...) ou mécaniques.

La solution adoptée dans le milieu industriel consiste à réduire au minimum les interactions entre l'homme et le robot. Des architectures adaptées de robots constituent des pistes de recherche dans les autres domaines : robot à structure molle dans le contexte de l'aide aux handicapés ; robot à sécurité passive dans le contexte des GMCAO.

- Les **imprécisions** lors de la réalisation de la tâche dues à des phénomènes d'usure, de jeux ou de calibrage inadapté.

Ces problèmes sont très peu pénalisants dans le cadre de l'aide aux handicapés. En revanche, une imprécision peut avoir des conséquences graves dans les domaines de l'industrie ou des GMCAO.

Dans la suite, nous allons nous attacher à vérifier que la tâche assignée au robot est réalisée de façon précise. Nous allons d'abord étudier les divers moyens de contrôle d'exécution envisageables.

Chapitre 2

Le contrôle d'exécution par capteurs de position externes

Au cours du chapitre précédent, nous avons distingué deux types d'erreur "matérielles" susceptibles de perturber l'utilisation d'un robot.

D'une part, il peut s'agir de grosses erreurs (codeur cassé, mauvaise transmission des données ...) qui engendrent des valeurs articulaires complètement absurdes. Ces erreurs peuvent être évitées en multipliant le matériel qui fournit les informations.

D'autre part, il peut s'agir de petites erreurs d'imprécision qui proviennent par exemple d'un mauvais calibrage du robot, d'une surcharge, de phénomènes d'usure ... Les informations articulaires sont exactes (codeurs en bon état et transmission correcte des valeurs fournies), mais le modèle du robot, insuffisamment précis, ne peut intégrer tous les paramètres, ce qui engendre une dérive des valeurs 3D de la position et de l'orientation de l'effecteur du robot.

Il faut donc un système externe, capable de renvoyer une information de position du robot, pour détecter ce type d'erreur. Le recours à un tel système externe de repérage du robot permet également de répondre au problème de la détection de grosses erreurs.

Nous avons tenté de répertorier les divers types de capteurs qui pourraient convenir pour concevoir un tel système de contrôle redondant. Nous avons alors envisagé deux solutions que nous comparons. Enfin, nous présentons les étapes de la solution retenue en précisant dans quelle partie du rapport elles seront développées.

2.1 Les divers types de capteurs envisageables

Nous divisons les capteurs 3D de position en deux catégories :

- ceux que nous appelons de type "digitaliseur" : ce sont des capteurs qui nous apprennent que le point de coordonnées (x, y, z) dans le repère du capteur, appartient à un objet de l'environnement. Ils reposent en général sur

l'interception d'un rayon (laser, lumineux ...) avec l'objet à détecter. Ils ne fournissent pas d'information sur le type de l'objet auquel appartient ce point. Ils ne permettent pas non plus de localiser un point donné d'un objet donné. En revanche, ils semblent bien adaptés à la simple détection de la présence d'obstacles.

- les capteurs qui permettent de détecter un point précis : ce point peut être une diode qui émet de la lumière infra-rouge, ou bien un émetteur ultra-sonique ... D'une manière générale, ce point est matérialisé par un émetteur, localisé par plusieurs récepteurs. Ce genre de capteurs fournit les coordonnées (x, y, z) (dans le repère du capteur) d'un point précis défini sur un objet. Ce point est précis dans le sens où sa position reste fixe par rapport à un repère associé à l'objet qu'on veut observer, et peut donc être définie dans ce repère.

Enfin, nous étudierons le cas particulier des caméras vidéo, qui selon la manière dont on les utilise, peuvent être assimilées à l'un ou l'autre type de capteurs.

2.1.1 Les capteurs de type “digitaliseur”

2.1.1.1 Les différentes techniques

Dans une première partie, nous décrirons les capteurs point à point, destinés à la numérisation 3D (les palpeurs mécaniques sont exclus du fait qu'ils ne présentent aucun intérêt pour notre étude). Puis une deuxième partie sera consacrée aux capteurs à mesure multiple, mieux adaptés à la numérisation rapide d'objets.

NB : Ce paragraphe s'inspire en grande partie de [Jah93]. Voir également [Woh92].

Les capteurs point à point

Nous allons examiner les différentes techniques permettant de localiser un point appartenant à un objet de l'environnement.

– Le temps de vol

Le principe est celui du radar dont sont équipées les chauves-souris : on envoie un signal (electromagnétique, optique ou ultrasonique) sur l'objet et on calcule la distance à partir du temps de vol de l'onde. L'équation de base du radar est : $vt = 2d$ où v est la vitesse de propagation du signal, d la distance à l'objet et t le temps d'aller-retour constaté.

Le bon fonctionnement d'un tel système dépend de la qualité de la réflexion de l'onde, elle-même tributaire de nombreux paramètres tels que l'angle d'incidence par rapport à la surface de l'objet, l'état de surface de l'objet ... Ces contraintes compliquent l'acquisition, la situation idéale étant un angle d'incidence de 90° sur une surface dispersant faiblement l'onde. Notons cependant qu'il existe des techniques de simulation de la réflexion d'une onde ultrasonore

sur un matériau donné, selon une incidence donnée. Ces techniques, basées sur une modélisation statistique de la réflexion du signal sonore sur des surfaces de matériaux différents, permettent d'apporter des corrections sur les résultats bruts fournis par les capteurs [ZSM93]. D'autre part, les systèmes qui exploitent le temps de vol le combinent souvent avec la triangulation. Ils sont ainsi plus robustes aux parties cachées.

Il existe plusieurs manières d'exploiter le temps de vol des ondes ultrasoniques ou electro-magnétiques. Les plus utilisées sont explicitées ci-dessous :

– Détection d'impulsion

Cette technique consiste simplement à envoyer une impulsion d'onde sur l'objet et à attendre l'écho. La distance se déduit directement du temps d'aller-retour.

L'avantage d'une telle méthode, outre sa simplicité, est son large domaine d'application : il n'existe pas de limite à la profondeur de champ et un radar assez puissant peut effectuer une mesure sur des objets grands et éloignés (dans le domaine de l'architecture par exemple).

– Modulation d'amplitude

Au lieu d'envoyer une impulsion brève, on peut moduler l'onde en amplitude. On mesure alors la différence de phase $\Delta\phi$ entre le signal transmis et le signal reçu. Cette différence de phase n'est déterminée que modulo 2π . La distance (fonction de $\Delta\phi$) ne peut donc être calculée que modulo un intervalle d'ambiguïté d_{ambig} équivalent à la longueur d'onde du signal émis. Dans ces conditions, la profondeur de champ est limitée. Une manière de s'affranchir de cette restriction consiste à combiner plusieurs fréquences de modulation.

La plupart des digitaliseurs radars commercialisés utilisent cette méthode plus robuste et plus rapide que la détection d'impulsion.

– **La triangulation active**

La triangulation se base uniquement sur la trigonométrie. Matériellement, elle utilise un rayon laser et une caméra. Lorsque le rayon laser intercepte un objet, il crée une tache ponctuelle, qui peut être observée par la caméra.

Sur la figure 2.1, la position et l'orientation du laser par rapport à la caméra sont parfaitement connues (b, θ) (pour simplifier la description, on a choisi le cas où le rayon laser est dans le plan d'équation $y = 0$). Le point (x, y, z) est projeté sur le plan image en (u, v) . Le théorème de Thalès nous fournit les égalités $u.z = x.f$ et $v.z = y.f$ et nous permet de déduire

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{b}{f \cdot \cot \theta - u} \begin{pmatrix} u \\ 0 \\ f \end{pmatrix}$$

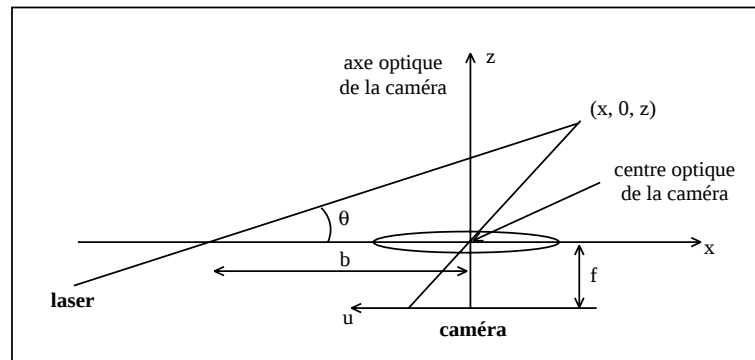


FIG. 2.1 - Principe de la triangulation active

Les parties cachées restent le problème inhérent aux méthodes par triangulation, puisque le point doit être vu à la fois par le laser et la caméra. L'utilisation de miroirs ou de plusieurs caméras diminue les risques d'occlusion respectivement du laser et des caméras.

– La mise au point

La technique consiste à utiliser les propriétés d'un modèle optique plus complet que le modèle à sténopé (modèle lentille mince ou modèle épais). L'image nette d'un point permet d'inférer des informations 3D en utilisant la relation de Descartes, classique en optique.

Ce processus nécessite une évaluation de la mise au point, au niveau du capteur photosensible (transformée de Fourier, ou plus simplement filtre passe-haut) [Kro87, Jar76, NN90].

L'inconvénient majeur de toutes ces techniques de point par point réside dans le fait que la réalisation d'une carte de profondeur dense nécessite des temps d'acquisition et de calcul importants. Aussi allons-nous explorer les mesures multiples.

Les capteurs à mesure multiple

Ils permettent d'acquérir "simultanément" la position de plusieurs points appartenant à un objet de l'environnement.

– Triangulation active : le plan laser

C'est une extension de la méthode décrite par la figure 2.1. Au lieu d'utiliser un seul rayon laser, on a recours à un plan laser (ce plan est créé soit par un miroir monté sur un galvanomètre, soit à l'aide d'une lentille cylindrique). Une ligne laser complète se trouve donc projetée sur l'objet au lieu d'un simple point.

– Triangulation active : le motif

Projeter une ligne laser sur l'objet permet d'obtenir plus de points. On peut aller plus loin, en projetant un motif combinant des structures simples (points et lignes). On peut même créer des structures qui discrétisent une grande partie de la surface de l'objet (l'une des plus simples étant une grille). Se pose alors le problème de l'identification de la trace laser, dont la solution générale consiste à coder les motifs. De nombreuses recherches ont été menées [SYI86, BK87, VO90]. Cependant peu de systèmes sont commercialisés. Les capteurs expérimentaux développés sont particulièrement rapides du point de vue de l'acquisition des données, mais nécessitent une grande puissance de calcul pour le traitement des données et l'obtention d'informations pertinentes à savoir les coordonnées 3D de points appartenant à la surface de l'objet observé.

– Le moiré

La technique par moiré consiste à illuminer l'objet à travers une grille régulière et à observer la scène à travers une grille identique par une caméra parallèle à la source. La figure d'interférence observée représente des lignes de niveau de l'objet, mais le signe de croissance ou décroissance de la profondeur est inconnu. On complète généralement la mesure par une autre après translation du capteur. Cette technique est performante pour des surfaces présentant peu de variation. On mesure énormément de points, jusqu'à 500 000, en une seule fois. L'exploitation des données recueillies reste le problème majeur de cette méthode, les lignes de niveau pouvant présenter de multiples discontinuités qu'il faut interpréter.

2.1.1.2 Comment les utiliser?

Ce qu'ils fournissent

Il est bien entendu que les capteurs à mesure multiple sont les plus intéressants puisqu'ils présentent des temps d'acquisition raisonnables. Ils fournissent un ensemble important de points $P_i = (x_i, y_i, z_i)$ appartenant à un objet de l'environnement.

Ces données seules ne permettent pas de déterminer de manière précise la position et l'orientation de l'objet observé, en l'occurrence le robot.

Ce dont on a besoin en plus

On a bien entendu envie de retrouver la pose¹ du robot à partir de ces informations. Pour cela, il faut mettre en correspondance cet ensemble de points avec

1. pose = position + orientation

Chapitre 2. Le contrôle d'exécution par capteurs de position externes

une autre représentation du robot. On pense naturellement à un modèle CAO. On a alors deux manières d'exploiter ces deux représentations du robot :

- On se sert de la connaissance a priori sur les valeurs articulaires du robot (valeurs renvoyées par les codeurs) : ces valeurs nous permettent d'obtenir un modèle CAO "rigide" du robot, correspondant à la configuration courante. On tente alors de le mettre en correspondance avec le nuage de points renvoyé par le digitaliseur. On mesure une erreur résiduelle. Si celle-ci est trop importante, la mise en correspondance n'a pas été possible. C'est donc que les valeurs articulaires fournies ne sont pas valables : on a détecté une erreur.
- On se sert de la connaissance a priori de manière moins directive : on ajoute les valeurs articulaires aux inconnues qui sont modifiées par la minimisation effectuée lors de la mise en correspondance (les valeurs renvoyées par les codeurs servent ici à initialiser ces valeurs articulaires inconnues). Ceci nous permet d'obtenir des valeurs articulaires que l'on peut comparer à celles fournies par le robot, ce qui permet de détecter une erreur.

Notons que l'on a besoin dans les deux cas d'un modèle CAO extrêmement précis. On peut cependant substituer à ce modèle CAO un modèle réalisé par une digitalisation préalable de chacune des parties du robot. Il faut ensuite assembler toutes ces parties afin de reconstituer un modèle cinématique du robot (cf paragraphe 2.2.1.1 2ème cas).

Avantages

Il est inutile de calculer la position précise du robot : il suffit de vérifier que sa forme externe est bien celle attendue.

Inconvénients

Quel digitaliseur utiliser ? Les techniques de moiré ou de triangulation active selon un motif, imposent un espace de travail restreint, ce qui est contraire aux contraintes de l'application nucléaire, et nécessitent des temps de calcul trop importants pour espérer obtenir un modèle complet du robot en "temps réel". L'acquisition de plusieurs plans laser serait plus rapide mais fournirait une information partielle.

Cette option peut cependant être retenue puisque des techniques de mise en correspondance avec des données partielles peuvent très bien fonctionner [BLS93].

J. Colly a ainsi développé un système de reconnaissance d'objets articulés à partir d'une carte de distances. Les données sont obtenues grâce à un télémètre laser monté sur une platine à deux degrés de liberté en rotation [Col91]. Ces informations de distance, complétées par les données d'un capteur de luminance permettent la reconnaissance d'objets polyédriques modélisés par un système de CAO.

2.1.2 Les capteurs permettant de détecter un point précis

Le principe général de ces capteurs est le suivant. On place un ou plusieurs émetteurs sur l'objet qui nous intéresse. Les émetteurs sont excités séquentiellement. Plusieurs récepteurs (au minimum 3) sont répartis tout autour de la région d'intérêt. La combinaison des informations fournies par les récepteurs permet de calculer la position 3D de l'émetteur courant dans un repère associé aux récepteurs.

2.1.2.1 Les différentes techniques

Les principales techniques utilisées sont :

Les systèmes électromagnétiques

Ils sont basés sur les champs magnétiques de basses fréquences. Chaque bobine génératrice du champ magnétique définit une direction spatiale précise. Les détecteurs de champs sont de petites bobines longitudinales.

Mais ces systèmes ne sont pas fiables : les champs magnétiques sont fortement perturbés par tout objet métallique environnant. Il existe bien sûr des modélisations de ces déformations (dont les paramètres peuvent être optimisés par des méthodes de calibrage). Cependant, elles nécessitent une connaissance complète de la géométrie et de la composition des objets susceptibles de perturber le système.

Dans le cas d'un robot, il nous faut donc connaître les informations précédentes pour chacune des parties du robot, ce qui implique de connaître leurs position et orientation respectives, c'est-à-dire les valeurs articulaires. Or ce sont justement les données qu'on veut vérifier. De plus, la quantité de calculs nécessaires à de telles corrections est colossale. Ce défaut est donc rédhibitoire pour l'utilisation d'un tel système appliqué à un robot.

Les systèmes ultrasoniques

La célérité du son dans l'air est assez lente pour pouvoir mesurer le temps de vol d'une onde ultrasonore entre un émetteur et un récepteur ultrasonores. Les distances entre un émetteur et chacun des récepteurs présents peuvent donc être estimées. La combinaison de ces distances permet de localiser l'émetteur par rapport à un repère 3D lié aux récepteurs.

Les fluctuations de temps de transit dues aux variations de la température, de la composition et du mouvement de l'air ne sont pas négligeables bien qu'elles puissent être minimisées par un calibrage préalable.

Les systèmes optiques

Ils sont basés sur l'émission d'un signal lumineux, infrarouge le plus souvent, qui est enregistré par plusieurs caméras : au moins deux caméras bi-dimensionnelles ou

au moins trois caméras linéaires. Ce sont des systèmes relativement précis. Northern Digital annonce pour son Optotrak une résolution de 0.01 mm à une distance de 2.5 m et prétend que l'erreur commise sur la mesure est de 0.15 mm environ. La pratique montre que cet ordre de grandeur est respecté, mais seulement pour un volume restreint ($0.7 \times 0.7\text{ m}^2$ à 2.5 m).

Les systèmes de suivi d'un rayon laser²

On peut également inclure dans la catégorie "détecteurs d'un point précis" les systèmes de "tracking laser". Ils diffèrent cependant des systèmes précédents du fait qu'il faut connaître la position initiale du point à suivre (on verra pourquoi ultérieurement) et qu'il ne faut pas perdre ce point en cours de traitement puisqu'il s'agit d'un suivi. Ce genre de systèmes pourrait néanmoins répondre à nos besoins.

Le principe de fonctionnement est illustré sur la figure 2.2.

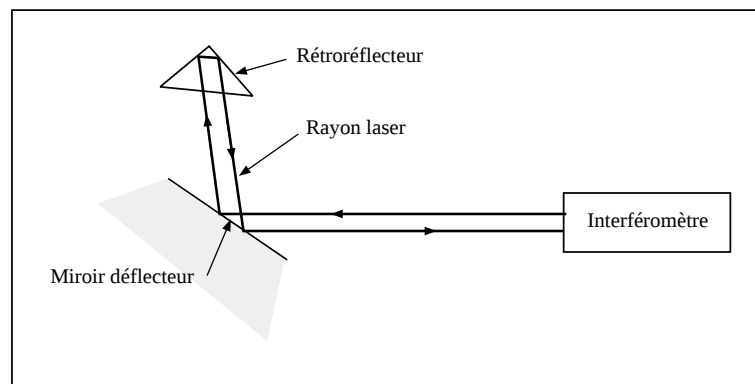


FIG. 2.2 - Principe du "tracking laser"

Le rétro-réflecteur est constitué de 3 miroirs perpendiculaires, ce qui implique que tout rayon qui l'intercepte ressort selon une direction parallèle. La position du miroir défecteur est commandable selon deux degrés de liberté en rotation (θ , ϕ). L'interféromètre fournit la distance parcourue par le rayon laser. Lorsque le miroir défecteur est parfaitement orienté, les chemins aller et retour du rayon laser sont confondus et le seul point du rétro-réflecteur atteint par le laser est l'intersection des trois miroirs perpendiculaires. On peut donc calculer la position de ce point à partir de la direction du rayon fournie par (θ , ϕ) et de la distance parcourue (l'interféromètre et le miroir défecteur doivent bien sûr être calibrés l'un par rapport à l'autre). Ainsi, l'origine du rétro-réflecteur, c'est-à-dire intersection des trois miroirs, est une sorte d'émetteur (dans le sens où c'est le point que l'on suit), mais qui a la particularité et l'avantage d'être passif. En revanche, il faut connaître sa position initiale pour orienter le miroir défecteur de telle sorte que le rayon laser intercepte le rétro-réflecteur.

2. Le terme anglais couramment employé est "tracking laser".

Le système proposé par [PVG93] est d'autant plus intéressant qu'il fournit non seulement la position, mais aussi l'orientation de l'effecteur du robot sur lequel a été fixé un rétroreflecteur. Il permettrait donc de suivre précisément l'effecteur du robot au cours de son déplacement.

Un tel système semble adapté à nos besoins, dans le sens où il est capable de suivre la position et l'orientation de l'effecteur du robot en temps réel. Mais, ce système impose néanmoins plusieurs restrictions :

- L'espace qui peut être surveillé est limité : le rayon laser doit être dans l'octant défini par les trois miroirs perpendiculaires constituant le rétroreflecteur. Cependant, il semble tout-à-fait envisageable de multiplier les rétroreflecteurs et les systèmes d'émission-réception du rayon laser (y compris l'interféromètre, la caméra CCD et le miroir de déflexion) afin de traquer l'effecteur du robot quelle que soit sa position et son orientation, tout en s'affranchissant des problèmes d'occlusion. Notons cependant que le coût de ce genre de capteur n'est pas négligeable du fait de sa complexité.
- Il serait indispensable de mettre au point une procédure permettant de calibrer le capteur extrinsèquement par rapport au bol.
- Enfin, l'encombrement de l'espace par un tel équipement semble un problème difficile à surmonter. En effet chaque système comporte un interféromètre, une caméra CCD, une diode sensible en position et un miroir deflecteur monté sur une articulation orientable de grande précision. L'utilisation de plusieurs de ces systèmes augmente d'autant l'encombrement.

2.1.2.2 Comment les utiliser ?

Ce qu'ils fournissent

Ils fournissent un ensemble restreint de points $P_i = (x_i, y_i, z_i)$. Chaque point correspond à un émetteur et est distinguable des autres points. En effet les émetteurs peuvent être stimulés successivement et on peut donc savoir lequel est actif dans le cas des systèmes électromagnétiques, ultrasoniques et optiques. Dans le cas du "tracking laser", il s'agit d'un suivi. Il suffit alors d'initialiser correctement le système pour distinguer les "émetteurs" tout le long de la surveillance.

On dispose donc d'informations ponctuelles que l'on peut regrouper pour constituer des repères 3D.

Ce dont on a besoin en plus

- une procédure pour constituer des repères à partir de plusieurs points. Ceci n'est pas indispensable, mais peut apporter un plus dans la manipulation des données

Chapitre 2. Le contrôle d'exécution par capteurs de position externes

- une procédure pour localiser un point (i.e. un émetteur) ou un repère (i.e. un groupe d'émetteurs) dans un repère associé à l'objet qui le porte (en l'occurrence l'une des parties du robot)
- une procédure de calibrage extrinsèque pour connaître la transformation entre le repère du robot (ou un repère absolu) et le repère du capteur. Cette procédure peut également s'avérer utile dans le cas des digitaliseurs, mais elle n'est pas indispensable s'il s'agit simplement de détecter une erreur dans les valeurs renvoyées par les codeurs (selon les méthodes décrites au paragraphe 2.1.1.2, il suffit de vérifier que la forme extérieure du robot obtenue par le digitaliseur correspond à la forme extérieure du robot prévue par un modèle CAO ou autre).

Comment les utiliser ?

A partir des valeurs articulaires renvoyées par le robot, on calcule son attitude. Grâce à la procédure de localisation d'un émetteur dans un repère associé à la partie du robot qui le porte (deuxième point de "ce dont on a besoin en plus"), on prévoit la position 3D de l'émetteur dans le repère associé à la base du robot.

On effectue alors la mesure de la position de l'émetteur dans le repère du capteur. On peut ensuite comparer ces deux valeurs (d'où la nécessité d'une procédure de calibrage extrinsèque du capteur dans le repère associé à la base du robot).

Avantages

La détection d'erreur est une simple comparaison de la position d'un certain nombre de points, c'est-à-dire le calcul de distance entre la position prévue de l'émetteur et sa position fournie par le capteur, au lieu d'être une minimisation longue entre deux ensembles de points dans le cas d'un capteur de type "digitaliseur" (cf paragraphe 2.1.1.2).

Inconvénients

Les localisateurs magnétiques subissent des perturbations à proximité d'une masse métallique. Ils sont donc inutilisables en présence d'un robot.

Les localisateurs ultra-soniques sont sensibles à la température et à la composition de l'air. Ils ne conviendraient donc pas à l'environnement de l'application nucléaire.

Les capteurs optiques du type de l'Optotrak sont très onéreux et il n'est pas envisageable de les utiliser en routine dans une salle d'opération par exemple. D'autre part, l'encombrement de ce genre de système interdit son utilisation dans une application telle que celle de l'EDF pour laquelle l'environnement est confiné (quart de sphère de 1,80 m de rayon).

Le même genre de restrictions s'applique aux systèmes de "tracking laser", ce qui est regrettable car ces systèmes présentent l'avantage d'utiliser des "émetteurs" passifs, qui n'ont donc pas besoin d'être alimentés et ne risquent pas de tomber en panne.

2.1.3 Le cas particulier des caméras vidéo

Selon la manière de les utiliser, elles peuvent être assimilées à des capteurs de type digitaliseur, ou bien elles permettent de connaître la position d'un point précis.

2.1.3.1 Utilisées en mode "digitaliseur"

Grâce à un traitement d'image approprié, on détecte les contours du robot (ou de tout objet qui nous intéresse). Par rétroprojection³ (possible après calibrage des caméras), on obtient pour chaque caméra, un cône dans l'espace, tangent à l'objet visualisé. L'utilisation simultanée de plusieurs caméras ne nous permet pas d'obtenir des points 3D. En effet, les contours détectés selon deux points de vue différents ne correspondent pas forcément aux mêmes parties de l'objet observé (ça peut malgré tout être le cas pour des objets polyédriques).

Malgré tout, ces données sont pertinentes dans l'optique d'une mise en correspondance. En effet, on peut procéder à une mise en correspondance 3D/2D [LSB91] au lieu d'une mise en correspondance 3D/3D, c'est-à-dire qu'on cherche à minimiser la distance entre un nuage de points (les points du modèle) et un nuage de droites (les droites de rétroprojection), au lieu de minimiser la distance entre deux nuages de points (les deux nuages de points étant les points du modèle et les points obtenus par le digitaliseur dans le cas d'un digitaliseur "classique").

Cependant, le traitement d'image nécessaire à la détection des contours peut s'avérer délicat : on détecte trop d'indices (ombres, autres objets ...) ou pas assez (manque de contraste). On n'est jamais sûr des indices trouvés dans l'image. Certes la procédure de minimisation peut tenir compte de ces incertitudes en accordant des poids plus ou moins forts aux points de contours et en éliminant certains points aberrants, mais il est difficile d'acquérir une certitude sur l'appartenance ou non de certains points à l'objet qui nous intéresse, sans le recours à des connaissances a priori.

Une manière d'utiliser ces connaissances a priori consisterait à projeter les contours du robot dans les images⁴ et à en déduire quels sont les indices détectés dans les images, susceptibles d'appartenir réellement aux contours. Cependant si, le robot étant mal positionné, une arête se projette à l'endroit où aurait dû se projeter une autre arête, la détection d'erreur n'est pas immédiate. C'est l'examen complet de tous les contours qui nous permettra de juger si le robot est en bonne position ou non.

3. voir l'introduction du chapitre 3 pour une définition de la rétroprojection

4. La projection est possible après calibrage des caméras

Chapitre 2. Le contrôle d'exécution par capteurs de position externes

Ce genre de méthodes a été utilisé par [Low92]. Celui-ci travaille sur des segments de droites détectés dans une image (extraite d'une série d'images). Il explore un arbre de recherche pour estimer quel segment de droite détecté doit être mis en correspondance avec quelle arête du modèle du robot. Il intègre des mesures de probabilité sur ces appariements afin d'examiner en priorité les mises en correspondance ayant le plus de chance d'aboutir, et donc d'éviter au maximum les retours en arrière dans l'exploration de l'arbre. Les images précédentes sont utilisées pour évaluer les probabilités des appariements dans l'image courante. Les résultats annoncés sont le traitement de 3 à 5 images par seconde, ce qui conditionne la vitesse de déplacement maximale de l'objet observé.

L'utilisation des résultats sur les images précédentes est un bon moyen de réduire le traitement de l'image courante. Cependant il reste les problèmes d'initialisation du processus et les résultats annoncés par Lowe ont été obtenus pour des objets plus simples qu'un robot. L'analyse de "scènes naturelles" (c'est-à-dire dans un environnement réel) n'est pas aisée et peut rarement permettre une réponse de type tout ou rien.

2.1.3.2 Utilisées en mode "détecteur d'un point précis"

Les caméras jouent le rôle des récepteurs. Reste à trouver un émetteur ponctuel approprié. On peut imaginer des motifs tracés sur le robot et reconnaissables de manière indubitable⁵.

Le traitement d'image pour localiser ces marqueurs est aisé du fait qu'il s'agit de reconnaître une forme bien particulière, que nous nous efforcerons de choisir la plus simple possible. De plus comme les marqueurs sont distinguables les uns des autres, la procédure pourra fournir une réponse de type tout ou rien. A chaque marqueur reconnu par une caméra, on associe une droite de rétroprojection (possible après calibrage de la caméra). Si le marqueur est vu par plusieurs caméras, l'intersection, au sens des moindres carrés, de toutes ces droites est la position 3D du marqueur en question.

Notons cependant que cette reconstruction 3D n'est pas indispensable pour vérifier que le marqueur est bien là où on l'attend. On peut se contenter de vérifier que les informations sont cohérentes en 2D dans chacune des images disponibles.

Le traitement d'image peut là aussi être simplifié par la connaissance a priori qui permet de prévoir où le marqueur doit apparaître dans l'image.

2.1.3.3 Avantages

En utilisant les caméras de cette façon, on a reconstitué exactement un capteur 3D de repérage d'un point précis. L'intérêt non négligeable de cette méthode est

5. Rappelons que les émetteurs des capteurs 3D décrits au paragraphe 2.1.1 étaient actifs et stimulés successivement afin de les distinguer les uns des autres. Dans le même but, on pourra utiliser plusieurs motifs (que l'on nommera désormais "marqueurs") différenciables les uns des autres selon le principe d'un code barre (par exemple).

que l'émetteur est passif. Il n'a donc pas besoin d'être alimenté et ne risque pas de tomber en panne.

2.1.3.4 Inconvénients

L'inconvénient le plus flagrant dans l'utilisation des caméras est le traitement d'images indispensable à l'extraction de l'information. Cependant il peut être facilité dans le cas de formes bien précises à détecter (marqueurs conçus par nos soins) dont on peut prévoir la position dans les images. Le choix du matériau employé pour concevoir ces marqueurs est également primordial. Plus il sera contrasté avec l'environnement, plus le traitement d'image sera simplifié.

2.1.3.5 Conclusion

Si l'on doit utiliser des caméras, il est plus judicieux de les utiliser en tant que détecteur d'un point précis puisque la partie délicate (à savoir le traitement d'image) est grandement simplifiée par la connaissance a priori dont on dispose (forme simple de plusieurs marqueurs très localisé plutôt que contours plus ou moins complexes du robot complet).

2.2 Comparaison des deux solutions envisageables

2.2.1 Récapitulatif des deux solutions

2.2.1.1 Mise en correspondance du robot digitalisé et de son modèle (modèle CAO ou modèle construit au préalable)

1er cas : On dispose d'un modèle CAO très précis du robot. Alors on instancie le modèle CAO avec les valeurs articulaires (q_i), $i = \{0..6\}$ renvoyées par le robot.

2ème cas : On n'a pas de modèle CAO assez précis.

Au cours de cette thèse, nous allons faire à plusieurs reprises, référence au localisateur Optotrak (voir Annexe A pour une description de son fonctionnement). Pour construire un modèle convenable, on peut utiliser un tel localisateur en procédant grossièrement de la manière suivante :

- On vient palper avec un palpeur Optotrak la surface du segment porteur du “rigid body” courant (voir Annexe A pour la définition d'un “rigid body”). On acquiert ainsi une série de points dans le repère associé au “rigid body”, ce qui permet de construire un modèle du segment courant (au moyen de fonctions splines par exemple).
- Il faut ensuite articuler ces différents segments. Pour cela, on reconstruit un modèle de Denavit-Hartenberg en identifiant les axes de rotation de chaque articulation. Considérons deux segments consécutifs. Si on les fait bouger l'un

par rapport à l'autre, l'ensemble des points exprimés dans $R_{rigidbody(i+1)}$ invariant dans $R_{rigidbody(i)}$ constitue l'axe de rotation recherché. A partir des différents axes de rotation, on peut reconstruire les repères $R_{seg(i)}$. Dans le cadre de l'utilisation de l'Optotrak comme digitaliseur, on a choisi un repère $R_{rigidbody}$ qui est matérialisé par une plaquette optotrak, mais quel que soit le digitaliseur utilisé, on peut construire un repère à partir de l'ensemble des points obtenus par digitalisation (par exemple par une décomposition selon les axes principaux [Hal79]).

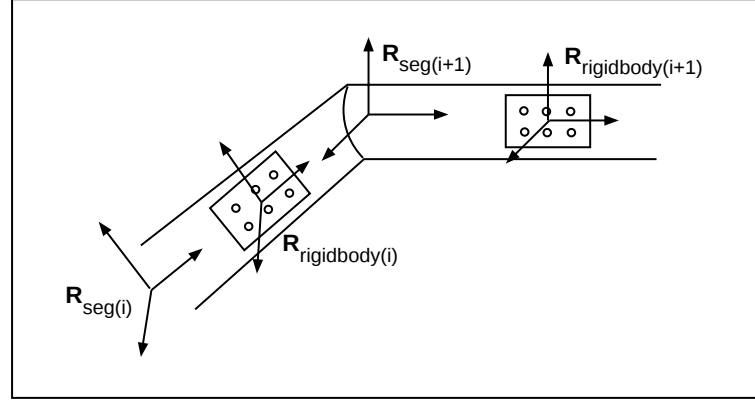


FIG. 2.3 - *Création d'un modèle du robot : on digitalise chaque segment seg_i par rapport à $R_{Rigidbody(i)}$ puis on cherche l'axe de rotation de l'articulation en bougeant deux segments successifs i et $i + 1$ l'un par rapport à l'autre.*

On dispose alors d'un modèle du robot pour effectuer des vérifications avec les informations fournies en direct par un digitaliseur.

Dans le cas où le digitaliseur fournit directement des points 3D, on cherche à mettre en correspondance deux ensembles de points (ceux fournis par le modèle et ceux fournis en direct par le digitaliseur). Pour cela, on minimise la distance entre ces deux ensembles de points en modifiant les valeurs (q_i) , $i = \{0..6\}$ injectées dans le modèle, ainsi que les paramètres d'une translation et d'une rotation.

Si les valeurs (q_i) , $i = \{0..6\}$ finales sont très différentes des valeurs q_i initiales, c'est qu'on a détecté une erreur. Si cette différence est faible, on a simplement ajusté ces valeurs.

Dans le cas où le digitaliseur utilisé est constitué de caméras, la procédure est plus délicate. On effectue une mise en correspondance 3D/2D entre le modèle et les indices détectés dans les images. Les inconnues sont les valeurs articulaires du robot $((q_i)$, $i = \{0..6\}$ initialisées avec les valeurs renvoyées par les codeurs). On modifie ces valeurs pour maximiser la somme des gradients⁶ calculés le long des contours du modèle projeté dans les images.

6. Un gradient fort est le signe de la présence éventuelle d'une frontière d'objet.

De même que pour la méthode avec un digitaliseur classique, si les valeurs (q_i) , $i = \{0..6\}$ finales sont très différentes des valeurs (q_i) , $i = \{0..6\}$ initiales, c'est qu'on a détecté une grosse erreur. Si cette différence est faible, on a simplement ajusté ces valeurs.

La méthode utilisant un digitaliseur 3D est plus sûre dans la mesure où elle ne nécessite pas de traitement supplémentaire (traitement d'images). L'ambiguïté sur l'appartenance au robot de points détectés est moins grande, puisqu'on a une localisation 3D, et non pas 2D⁷. On peut donc aisément éliminer les points qui ne concernent pas le robot, lors de la minimisation. Le pourcentage de points aberrants⁸ ainsi éliminés nous donne une indication sur le bien fondé de la mise en correspondance.

Reste à trouver le digitaliseur adéquat qui respecte les contraintes imposées par son utilisation dans le bol d'un générateur de vapeur de l'EDF ou dans une salle d'opération. On a vu que les digitaliseurs de type "triangulation laser par des motifs" ou "moiré" nécessitent des temps de calcul importants. Ils seraient peut-être envisageables dans le cadre d'une salle d'opération où l'espace à surveiller est relativement restreint. En revanche, ils ne conviennent pas pour un environnement aussi important que celui de l'EDF. Il reste donc les capteurs de type "plans laser". Il faut disposer, d'un nombre suffisant de plans pour intersecter le robot plusieurs fois en chacun de ses segments quelle que soit sa position dans le bol. Il faut donc au moins une trentaine de plans laser pour couvrir tout le bol selon des directions différentes (pour digitaliser un côté et le côté opposé d'un segment du robot). On peut approximativement les positionner tous les 10 cm, selon 2 directions opposées. La hauteur du bol étant 1,80 m, on peut envisager 2 x 17 plans.

Les caméras doivent également être en nombre suffisant pour couvrir tout l'espace du bol, tout comme pour l'autre solution retenue. Ce n'est donc pas un critère discriminant dans le choix de l'une ou l'autre solution.

2.2.1.2 Surveillance de marqueurs dans des images vidéo

On place des marqueurs sur chacune des parties du robot. Ces marqueurs sont calibrés, c'est-à-dire qu'on connaît leur position - et éventuellement leur orientation - par rapport au repère associé au segment du robot qui le porte.

- Le robot renvoie les valeurs (q_i) , $i = \{0..6\}$.
- On prévoit la position 3D de chaque marqueur dans cette configuration du robot.
- On projette ces points dans chacune des images, ce qui est possible grâce au calibrage des caméras.

7. l'ambiguïté est en effet plus grande en 2D puisque la projection rapproche des points qui en fait sont éloignés dans l'espace.

8. Les points aberrants sont les points dont la distance au modèle est supérieure à un seuil donné.

Chapitre 2. Le contrôle d'exécution par capteurs de position externes

- On cherche dans les images le marqueur courant autour de l'endroit où il devrait apparaître, c'est-à-dire autour de sa projection.
- Si on ne le trouve pas, c'est qu'il y a une grosse erreur ou bien qu'il est occulté par un objet présent dans l'environnement. On peut distinguer ces 2 cas en fonction des résultats renvoyés par les autres caméras et les autres marqueurs.
- Si le marqueur est vu par plusieurs caméras, on le rétroprojette et l'on calcule sa position 3D par intersection (au sens des moindres carrés) des droites de rétroprojection. On compare la position 3D prévue et la position 3D recalculée. On évalue ainsi les petites erreurs.
- Si le marqueur n'est vu que par une seule caméra, on peut quand même en tirer une information intéressante en le rétroprojetant et en calculant la distance entre cette droite de rétroprojection et la position 3D prévue. Ceci nous permet également d'évaluer les petites erreurs. C'est d'ailleurs la manière de procéder la plus souple puisqu'elle permet d'exploiter toutes les informations fournies par les caméras.

L'utilisation des caméras, plutôt que tout autre capteur 3D permettant de localiser un point précis, est justifiée par le moindre coût du matériel impliqué et par le fait que les émetteurs sont passifs (on n'a donc pas besoin d'alimentation supplémentaire à proximité du robot, ce qui permet de l'instrumenter aisément).

2.2.2 Comparaison des deux méthodes

Les différentes étapes préliminaires au contrôle redondant sont primordiales, mais ce ne sont pas elles qui vont nous déterminer dans le choix de l'une ou l'autre des solutions. En effet, elles peuvent être réalisées de manière indépendante, et le sont généralement une fois pour toutes. Elles ne sont donc pas contraignantes vis-à-vis du processus de contrôle lui-même. Il s'agit des étapes telles que le calibrage intrinsèque des capteurs, la construction d'un modèle du robot (pour la première solution envisagée) ou le calibrage des marqueurs (pour la deuxième solution envisagée).

Les critères qui vont orienter notre choix sont liés à la surveillance proprement dite du robot et à la qualité des informations qu'on peut obtenir. Nous les synthétisons dans le tableau suivant.

Élément de comparaison	Méthode par mise en correspondance (1)	Méthode par surveillance de marqueurs (2)	“Vainqueur”
Taille de l'espace qu'il est possible de surveiller	Inutile de recalculer les observations par rapport à l'environnement (vérification uniquement de la forme externe du robot). Seule contrainte = calibrer les capteurs les uns par rapport aux autres (possible grâce à une mire visible par deux capteurs simultanément)	Il faut recalculer les résultats par rapport à l'environnement, ce qui pourrait réduire la taille de l'espace surveillé. Cependant, le recours à des mires visibles par deux caméras simultanément permet d'étendre le calibrage extrinsèque de proche en proche.	L'espace surveillé est donc identique pour les 2 méthodes et dépend uniquement du nombre et des caractéristiques des capteurs.
Type d'informations fournies par les capteurs	Ensemble de points appartenant à un objet présent dans l'environnement (information globale) $\Rightarrow$ ambiguïté sur l'appartenance des points au robot	Ensemble de points précis (repérés par rapport au robot) $\Rightarrow$ unicité des points assurée $\Rightarrow$ aucune ambiguïté	(2)+++
Type de détection d'erreur	On vérifie la cohérence sur tous les axes à la fois.	On examine la cohérence seulement à l'extrémité, mais c'est ce qui est primordial et permet d'en déduire la cohérence sur tous les axes.	
Temps de détection d'erreur	Calcul de la distance de chacun des points digitalisé au modèle $\Rightarrow$ détection lente, même avec un calcul préalable de carte de distances	Calcul de distance entre points prédits et points réellement détectés, sur un nombre limité de points $\Rightarrow$ détection rapide	(2)+++
Coût matériel	Un certain nombre de digitaliseurs (caméras + plans laser). Ce nombre dépend de l'espace à surveiller.	Un certain nombre de caméras. Ce nombre dépend de l'espace à surveiller.	(2)+

2.3 Notre approche : le contrôle redondant par suivi de marqueurs dans des images vidéo

Le tableau comparatif précédent nous incite à choisir la seconde solution qui consiste à surveiller des marqueurs. La détection d'erreur est plus sûre (cf type d'informations fournies par le capteur). Elle est plus rapide puisqu'elle compare seulement quelques points précis au lieu d'effectuer une minimisation globale sur un calcul de distance entre deux ensembles de points, ce qui permet d'envisager plus aisément un suivi temps réel.

2.3.1 Présentation de la méthode retenue

L'idée générale consiste à suivre plusieurs marqueurs placés sur l'effecteur du robot : on prévoit, en fonction de valeurs fournies par les codeurs, l'endroit où doivent apparaître ces marqueurs dans les images renvoyées par les caméras. Il suffit de vérifier que ces marqueurs sont à la bonne place pour s'assurer que le robot est bien dans la configuration annoncée par les codeurs.

En réalité, il est possible de positionner les marqueurs à un même endroit selon des configurations différentes (redondance des configurations). Néanmoins, pour que cela se produise, il faut qu'il y ait au moins deux valeurs codeurs modifiées et que toutes les valeurs codeurs modifiées le soient de manière à se compenser mutuellement. Ceci est tout-à-fait inenvisageable dans le cadre d'erreurs telles que des pannes codeurs ou des erreurs de communication.

Dans le cas de petites erreurs, telles qu'une dérive du robot, les erreurs de plusieurs articulations peuvent se compenser. Cependant, la déviation de chaque segment par rapport à sa position commandée n'est pas dangereuse si une marge de sécurité suffisante a été prévue lors de la planification de la trajectoire. Ce qui importe alors, c'est uniquement la précision du positionnement et de l'orientation de l'effecteur. C'est justement elle que nous nous proposons de vérifier par le contrôle redondant.

La détection d'une erreur de positionnement du robot est donc équivalente à la détection d'une erreur de positionnement d'au moins un marqueur.

Une fois une erreur détectée, pour en diagnostiquer l'origine, c'est-à-dire pour détecter quelle est l'articulation défectueuse, on peut envisager d'examiner les segments un à un, en partant de l'effecteur et en remontant jusqu'à la base du robot. Cette deuxième phase ne nécessite aucune contrainte de temps, étant donné qu'elle est effectuée après l'arrêt du robot.

2.3.2 Présentation du rapport : quels sont les problèmes à résoudre ?

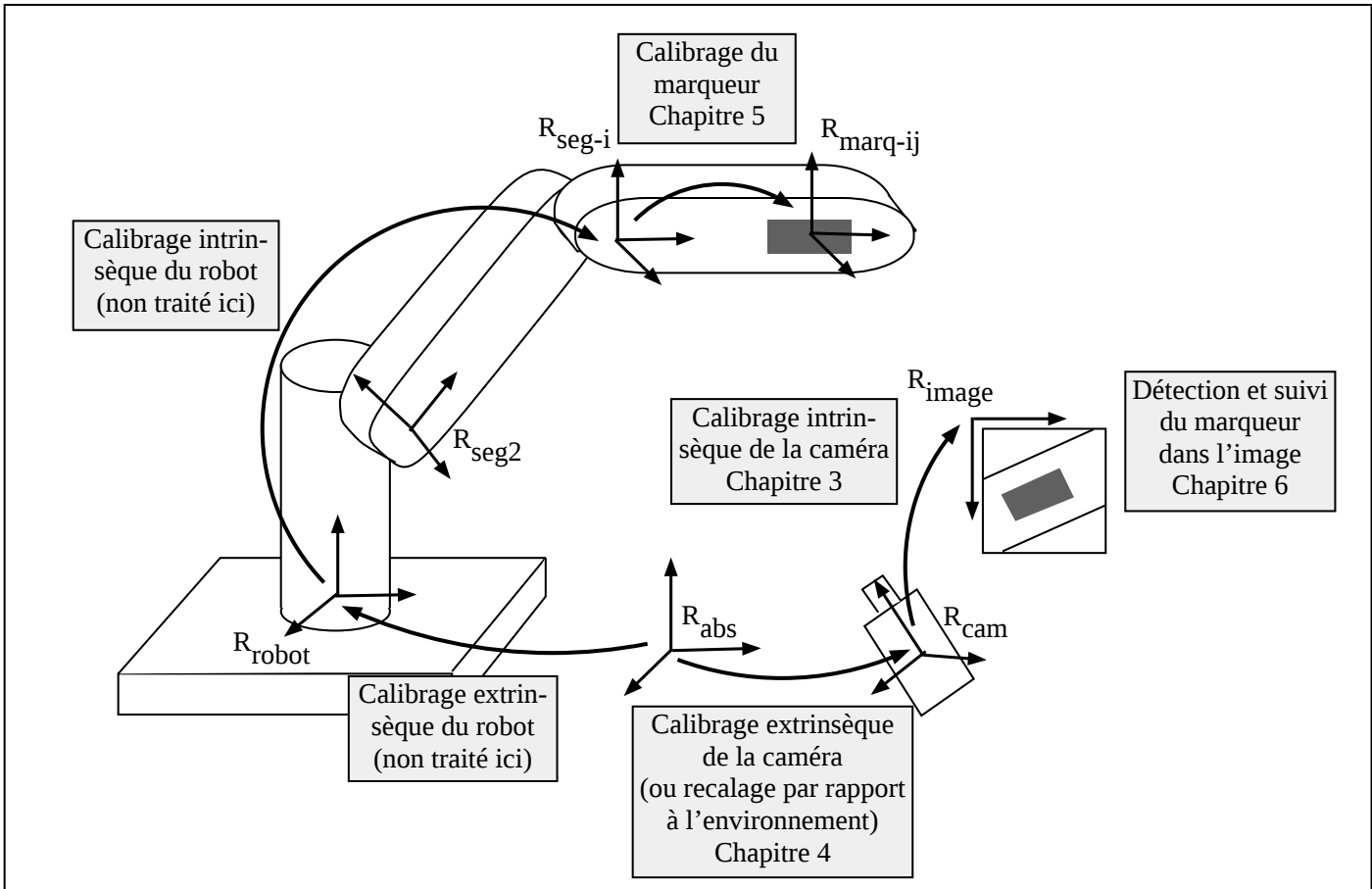


FIG. 2.4 - Présentation générale du projet

Chapitre 2. Le contrôle d'exécution par capteurs de position externes

Avant de décomposer le projet selon ses diverses phases, explicitons les notations employées pour désigner les différents repères (voir figure 2.4) :

- R_{abs} est le repère absolu. C'est un repère fixe associé soit à la salle d'opération, soit au bol du générateur de vapeur.
- R_{robot} est le repère associé à la base du robot.
- R_{seg-i} est le repère associé au $i^{ème}$ segment du robot. Il se déplace par rapport à R_{robot} en fonction des valeurs articulaires (q_k) $k = \{0, \dots, i-1\}$.
- $R_{marq-ij}$ est le repère associé au $j^{ème}$ marqueur du $i^{ème}$ segment.
- R_{cam} est un repère "abstrait" associé à la caméra.
- R_{image} est un repère 2D associé à l'image fournie par la caméra.

Les principales phases sont :

- **le calibrage intrinsèque du robot** : c'est la détermination des paramètres du robot qui vont permettre de construire la matrice de passage entre R_{robot} et R_{seg-i} en fonction des valeurs articulaires (q_k) $k = \{0, \dots, i-1\}$.

Nous considérons que cette phase a déjà été effectuée. Elle sera traitée de manière très succincte dans le chapitre concernant les expérimentations complètes (chapitre 7).

- **le calibrage extrinsèque du robot** : c'est la détermination de la matrice de passage entre R_{robot} et R_{abs} .

Cette phase est spécifique à chaque environnement. Lors des expérimentations, nous choisirons comme repère absolu, le repère associé à la base du robot. Cette phase n'aura donc pas lieu d'être.

- **le calibrage intrinsèque des caméras** : c'est la détermination d'une fonction qui à un point (u, v) défini dans R_{image} , associe une droite 3D définie dans R_{cam} . C'est également la détermination de la fonction inverse, c'est-à-dire d'une fonction de projection d'un point 3D défini dans R_{cam} , en un point 2D défini dans R_{image} .

Cette étape sera abordée dans le **chapitre 3**.

- **le recalage des caméras par rapport à l'environnement** (ou calibrage extrinsèque des caméras) : c'est la détermination de la matrice de passage entre R_{cam} et R_{abs} .

On développera cette phase au **chapitre 4**.

- **le calibrage des marqueurs** : c'est la détermination de la matrice de passage entre R_{seg-i} et $R_{marq-ij}$. On suppose que le robot est suffisamment rigide pour que cette matrice soit fixe au cours du temps et des variations de configurations.

Le calibrage des marqueurs sera détaillé au **chapitre 5**.

A ces phases préliminaires, il faut ajouter les actions exécutées au cours du processus de surveillance : **détection, identification et suivi des marqueurs**, problèmes qui seront traités au **chapitre 6**.

Enfin, on décrira une **expérimentation** complète du processus au **chapitre 7** avant d'examiner les **limitations** de cette approche, ainsi que ses **extensions** possibles au chapitre 8.

Chapitre 3

Calibrage intrinsèque des caméras

Les caméras sont utilisées comme des outils de mesure. Les informations recherchées sont les positions tri-dimensionnelles d'objets réels, alors que les informations fournies sont les images bi-dimensionnelles de ces objets.

Il est donc souhaitable d'exploiter les informations 2D pour les transformer en des informations 3D intéressantes. Le calibrage a pour objet de déterminer avec précision une telle transformation, appelée *rétroprojection* : à tout pixel¹ (u, v) , on veut associer la demi-droite constituée de l'ensemble des points 3D de l'espace dont l'image est (u, v) .

La transformation inverse, la *projection*, permet de prévoir l'image (u, v) d'un point 3D (x, y, z) à travers la caméra. Dans notre application, elle ne nécessite pas une grande précision puisqu'elle ne sert qu'à prédire grossièrement la position d'un marqueur dans l'image, en vue de restreindre la fenêtre de recherche lors de la phase de traitement d'image. Nous allons donc essentiellement nous intéresser à la *rétroprojection*.

3.1 Etude bibliographique

Le calibrage repose sur le fait qu'on connaît à la fois la position 3D d'un certain nombre d'indices et la position 2D de leurs images respectives. Les indices sont généralement des points, mais il peut s'agir d'autres formes géométriques telles que des droites [CT90, Vai90] ou des ellipses [TG94]. Par construction de la mire, on connaît les coordonnées des points 3D dans R_{mire} , un repère associé à la mire. Les coordonnées 2D des images de ces points sont obtenues dans R_i , un repère associé à l'image, grâce à un traitement d'image le plus précis possible. Nous étudierons des techniques de détection précise d'un point dans le paragraphe 6.2.2. Dans la suite de ce chapitre, nous supposerons que nous disposons de ces deux jeux de données. Ces informations permettent de déterminer les paramètres d'un modèle du dispositif d'acquisition d'images qui comprend l'objectif, la caméra et la carte de digitalisation. C'est ce dispositif complet que nous calibrons, et non pas seulement la caméra. Le

1. le terme pixel provient de l'anglais "PICture ELe ment" et signifie donc "élément de l'image".

changement d'un seul des éléments qui le composent nécessite un nouveau calibrage.

On classe les méthodes de calibrage selon le type de modélisation qu'elles utilisent. Le modèle à sténopé vise à reproduire le processus de formation théorique d'une image. Ses faiblesses ont conduit à le compléter par une modélisation physique ou mathématique des distorsions. Enfin les modèles multi-plans permettent une modélisation purement mathématique, qui intègre sans les modéliser explicitement le phénomène de création d'image, les distorsions optiques, les perturbations électroniques ...

3.1.1 Modèle à sténopé

Le sténopé est le plus simple des objectifs photographiques. Il consiste en un trou d'épingle percé dans la chambre noire de l'appareil photographique. Les rayons lumineux de la scène observée passent à travers ce trou et impressionnent le film placé au fond de la chambre noire.

3.1.1.1 Construction du modèle

La géométrie du sténopé est typiquement celle d'une projection centrale.

Si l'on voulait se conformer à la réalité physique du sténopé, le plan de projection serait placé derrière le centre de projection, mais la représentation équivalente couramment adoptée est celle de la figure 3.1.

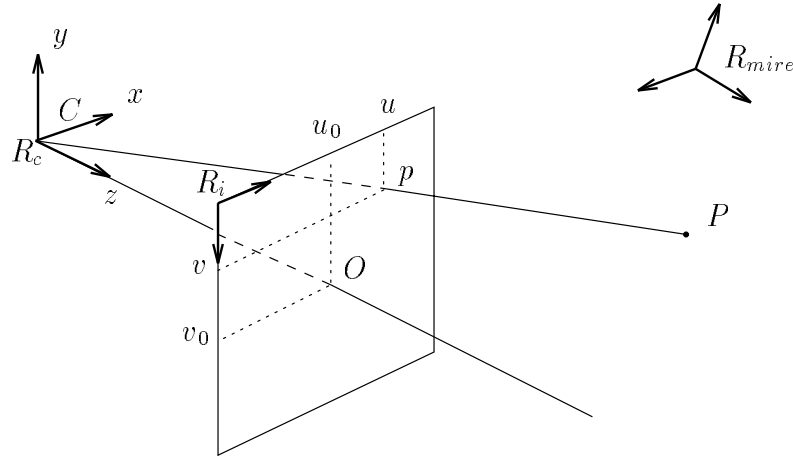


FIG. 3.1 - *Modèle à sténopé*

Le repère R_{mire} est le repère associé à la mire de calibrage. C'est dans ce repère que l'on connaît les coordonnées 3D des points de calibrage.

Le repère R_c associé à la caméra est un repère intermédiaire qui facilite l'expression du modèle : l'origine C est le centre de projection ; l'axe optique $[Cz)$ est perpendiculaire au plan de projection, c'est-à-dire à l'image ; les axes $[Cx)$ et $[Cy)$

sont parallèles respectivement aux lignes et aux colonnes de l'image. L'unité de mesure dans ce repère est le millimètre.

Le repère R_i est le repère 2D associé à l'image. L'unité de mesure dans ce repère est le pixel.

Soit P , un point quelconque de l'espace, observé par la caméra. Soient $(X_m, Y_m, Z_m)^t$ ses coordonnées dans le repère R_{mire} et $(X_c, Y_c, Z_c)^t$ ses coordonnées dans le repère R_c . La projection de P par la caméra est le point p . p a pour coordonnées $(x, y, F)^t$ dans le repère R_c , où F désigne la longueur focale du système², et (u, v) dans le repère R_i .

L'équation de la projection peut s'écrire de la manière suivante :

$$(x, y, F)^t = k.(X_c, Y_c, Z_c)^t \quad (3.1)$$

Le point O est l'intersection de l'axe optique et du plan de projection. Il a pour coordonnées $(0, 0, F)$ dans le repère R_c et (u_0, v_0) dans le repère R_i

Alors, les coordonnées du point projeté p , exprimées dans R_c , et celles du pixel correspondant, exprimées dans le repère de l'image R_i , sont reliées par les facteurs d'échelle positifs k_u et k_v :

$$\begin{aligned} x &= k_u(u - u_0) \\ y &= -k_v(v - v_0) \\ z &= F \end{aligned} \quad (3.2)$$

Si l'on essaie de combiner les équations 3.1 et 3.2 et d'éliminer le facteur k , on voit que les coordonnées d'un pixel (u, v) sont des fonctions non linéaires des variables X_c, Y_c et Z_c . Cependant, l'utilisation des coordonnées homogènes permet de formuler une relation linéaire :

- soit $P_c = (X_c, Y_c, Z_c, 1)^t$ le vecteur des coordonnées homogènes de P définies dans le repère de la caméra R_c .
- soit P_i le vecteur $(su, sv, s)^t$ des coordonnées homogènes du pixel p correspondant, u et v étant définies dans le repère R_i .

Les équations 3.1 et 3.2 permettent alors d'écrire :

$$P_i = T_{int}.P_c \quad \text{avec} \quad T_{int} = \begin{pmatrix} \frac{F}{k_u} & 0 & u_0 & 0 \\ 0 & \frac{-F}{k_v} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & -\alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Les 4 paramètres α_u, α_v, u_0 et v_0 de la matrice T_{int} sont dits paramètres intrinsèques du modèle. Ils définissent le fonctionnement interne de la caméra et ne tiennent pas compte de sa position par rapport à la mire de calibrage.

2. La longueur focale est la distance entre le centre de projection et le plan de projection. C'est principalement elle qui détermine l'agrandissement de l'image.

Chapitre 3. Calibrage intrinsèque des caméras

Ce sont les paramètres extrinsèques qui intègrent cette information. Il s'agit de 3 translations et de 3 rotations qui constituent la matrice de passage T_{ext} entre le repère interne de la caméra R_c et le repère externe R_{mire} associé à la mire de calibrage et dans lequel sont connues les coordonnées des points de référence. On a :

$$P_c = T_{ext}.P_m \quad \text{avec} \quad T_{ext} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{D'où} \quad P_i = T_{int}.T_{ext}.P_m = M.P_m$$

$$\text{c'est-à-dire} \quad \begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.3)$$

Notons que pour résoudre certains problèmes qui deviennent quadratiques à cause du facteur d'échelle s , on peut avoir recours à des modèles de caméras plus simple que le modèle à sténopé. Il s'agit des modèles affines, qui imposent à s la valeur 1. Ce sont des approximations du modèle à sténopé. On peut citer le modèle orthographique, qui est le plus grossier, le modèle orthographique à l'échelle qui est une approximation à l'ordre 0 et le modèle paraperspectif qui est une approximation à l'ordre 1 [Mor95]. Dans la suite, nous n'étudierons pas ces modèles qui sont commodes mais supposent que les points de calibrage et les points observés ensuite sont tous très proches d'un plan $\mathcal{P}$ perpendiculaire à l'axe optique de la caméra.

3.1.1.2 Détermination des paramètres

Pour calibrer la caméra, il faut d'abord estimer la matrice M , puis extraire les paramètres intrinsèques (les seuls qui nous intéressent réellement ici) à partir de ses coefficients m_{ij} :

- L'estimation de la matrice M est possible par la résolution d'un système linéaire d'équations :

L'équation 3.3 peut être réécrite sous la forme :

$$\begin{aligned} m_{11}X + m_{12}Y + m_{13}Z + m_{14} - u.(m_{31}X + m_{32}Y + m_{33}Z + m_{34}) &= 0 \\ m_{21}X + m_{22}Y + m_{23}Z + m_{24} - v.(m_{31}X + m_{32}Y + m_{33}Z + m_{34}) &= 0 \end{aligned}$$

On appelle $i^{ème}$ mesure, la connaissance des coordonnées 3D du $i^{ème}$ point de calibrage $P_i = (X_i, Y_i, Z_i)^t$ dans R_{mire} et des coordonnées 2D (u_i, v_i) de

son image dans R_i . Donc si l'on dispose de n mesures, on a $2n$ combinaisons linéaires des coefficients m_{ij} :

$$\begin{pmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_i X_i & -u_i Y_i & -u_i Z_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_i X_i & -v_i Y_i & -v_i Z_i & -v_i \\ & & & & & & & & & & & \vdots \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{pmatrix} \vdots \\ 0 \\ 0 \\ \vdots \end{pmatrix}$$

que l'on peut écrire plus synthétiquement sous la forme : $A_{12} \cdot \vec{m}_{12} = \vec{0}$. Si l'on dispose d'un nombre suffisant de données (chaque point fournissant 2 équations, il faut au moins 6 points pour déterminer les 12 coefficients de la matrice M), on peut déterminer les m_{ij} à un coefficient multiplicatif près. La contrainte la plus simple à mettre en oeuvre pour s'affranchir de ce coefficient multiplicatif consiste à imposer $m_{34} = 1$.

Le système d'équations précédent est alors équivalent au système $A_{11} \cdot \vec{m}_{11} = \vec{b}$, où A_{11} est la matrice constitué des 11 premières colonnes de A_{12} , $\vec{m}_{11}$ est le vecteur inconnu constitué des 11 premiers éléments de $\vec{m}_{12}$ et $\vec{b} = (\dots u_i v_i \dots)^t$.

On peut alors résoudre le système d'équations précédent par l'utilisation de la pseudo-inverse : $\vec{m}_{11} = (A_{11}^t A_{11})^{-1} A_{11}^t \vec{b}$ (voir [PFTV92], pages 528-539).

Faugeras et Toscani [FT87] proposent d'utiliser la contrainte $m_{33}^2 + m_{32}^2 + m_{31}^2 = 1$ qui provient du fait que M est le produit des matrices T_{int} et T_{ext} , ce dont on déduit : $m_{31} = r_{31}$, $m_{32} = r_{32}$ et $m_{33} = r_{33}$. Or, r_{31}, r_{32} et r_{33} sont les coefficients d'une ligne d'une matrice de rotation. Donc la somme de leurs carrés vaut 1.

- Les paramètres intrinsèques et extrinsèques peuvent être extraits en identifiant chacun des coefficients de la matrice M précédemment estimée avec l'expression algébrique du coefficient correspondant dans le produit de matrices $T_{int} \cdot T_{ext}$. Il faut également tenir compte des propriétés d'orthonormalité de la rotation et imposer que α_u et α_v soient positifs. En posant $\vec{m}_i = (m_{i1} m_{i2} m_{i3})^t$ et $\vec{r}_i = (r_{i1} r_{i2} r_{i3})^t$, on a :

$$\begin{aligned} \vec{r}_3 &= \vec{m}_3 \\ t_z &= m_{34} \\ u_0 &= \vec{m}_1 \cdot \vec{m}_3 \end{aligned}$$

Chapitre 3. Calibrage intrinsèque des caméras

$$\begin{aligned}
v_0 &= \vec{m}_2 \cdot \vec{m}_3 \\
\alpha_u &= \|\vec{m}_1 \wedge \vec{m}_3\| \\
\alpha_v &= \|\vec{m}_2 \wedge \vec{m}_3\| \\
\vec{r}_1 &= (\vec{m}_1 - u_0 \vec{m}_3) / \alpha_u \\
\vec{r}_2 &= -(\vec{m}_2 - v_0 \vec{m}_3) / \alpha_v \\
t_x &= (m_{14} - u_0 m_{34}) / \alpha_u \\
t_y &= -(m_{24} - v_0 m_{34}) / \alpha_v
\end{aligned}$$

3.1.1.3 Conclusion

Le modèle à sténopé conserve l'alignement des points et agit de la manière suivante : l'image d'un point est l'intersection du plan image et de la droite passant par ce point et par le centre de projection ; la droite de rétroprojection d'un pixel est la droite passant par ce pixel et le centre de projection. Ceci permet de l'utiliser facilement dans un sens ou dans l'autre (projection ou rétroprojection).

Cependant ce modèle simple ne tient pas compte des imperfections du système de prise de vue qui engendrent des distorsions.

3.1.2 Intégration des distorsions dans le modèle à sténopé

Les distorsions sont dues à des lentilles imparfaites, à des défauts mécaniques de montage, mais aussi à la digitalisation du signal vidéo.

$$\text{Ainsi, au lieu d'avoir : } \begin{cases} u &= u_0 + \alpha_u \frac{X_c}{Z_c} \\ v &= v_0 + \alpha_v \frac{Y_c}{Z_c} \end{cases}$$

$$\text{on a : } \begin{cases} u' &= u + \Delta_u = u_0 + \alpha_u \frac{X_c}{Z_c} + \Delta_u \\ v' &= v + \Delta_v = v_0 + \alpha_v \frac{Y_c}{Z_c} + \Delta_v \end{cases}$$

(u, v) sont les coordonnées idéales du pixel correspondant au point 3D (X_c, Y_c, Z_c) . Ce sont les coordonnées qu'on devrait obtenir si la caméra pouvait être modélisée comme un sténopé.

(u', v') sont les coordonnées effectives du pixel correspondant au point 3D (X_c, Y_c, Z_c) .

Les distorsions (ou corrections) Δ_u et Δ_v peuvent être modélisées de différentes manières.

3.1.2.1 Modélisation physique globale

Seules les distorsions optiques (défauts des lentilles elle-mêmes et/ou de leur montage mécanique) peuvent être modélisées physiquement. On en recense trois

types : distorsions radiale (les lentilles présentent de légères courbures), de parallélisme (erreurs de parallélisme entre les différentes lentilles), de décentrage (par rapport à l'axe optique). La distorsion radiale est la plus influente [TG94] et c'est généralement la seule considérée. Elle est de la forme :

$$\begin{cases} \Delta_u &= k r^2 (u - u_0) \\ \Delta_v &= k r^2 (v - v_0) \end{cases} \quad \text{avec} \quad r^2 = (u - u_0)^2 + (v - v_0)^2$$

Il faut donc déterminer un paramètre supplémentaire, k . Mais la tâche n'est pas aisée car on a perdu la linéarité du problème. On ne peut donc pas obtenir une solution analytique et le recours à des méthodes itératives s'avère nécessaire. Il s'agit de déterminer l'ensemble de paramètres qui minimise dans l'image, la différence entre les données réelles (coordonnées des points extraits de l'image) et les valeurs obtenues après avoir appliqué le modèle aux points 3D de la mire. La valeur à minimiser est donc :

$$S = \sum_{i=1}^N d^2 \left(\begin{pmatrix} u_i \\ v_i \end{pmatrix}, M_p \cdot \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \right)$$

où $\begin{cases} N \text{ est le nombre de points de calibrage} \\ d \text{ est la distance euclidienne 2D} \\ (X_i \ Y_i \ Z_i)^t \text{ est le } i^{\text{ème}} \text{ point de calibrage 3D} \\ M_p \text{ est le modèle de formation de l'image qui dépend de l'ensemble } p \text{ des} \\ \text{paramètres à optimiser } (p = \{u_0, v_0, \alpha_u, \alpha_v, k, R, t\}) \\ (u_i \ v_i)^t \text{ sont les coordonnées du } i^{\text{ème}} \text{ point calculées dans l'image} \end{cases}$

Notons que l'on peut travailler dans l'espace et chercher à minimiser la somme au carré des distances entre le $i^{\text{ème}}$ point 3D $(X_i \ Y_i \ Z_i)^t$ et la droite de rétroprojection associée à la $i^{\text{ème}}$ image détectée $(u_i \ v_i)^t$.

Pour que la minimisation itérative soit efficace, il faut avoir une bonne initialisation des paramètres. On applique généralement la méthode linéaire précédemment décrite (ou l'une de ses variantes) pour déterminer une bonne approximation de tous les paramètres excepté k . Puis l'on injecte ce résultat dans le processus itératif.

On a alors recours à des techniques d'optimisation non linéaire telles que les algorithmes de Levenberg-Marquardt ou de Gauss-Newton ([PFTV92] chapitres 10 et 14) .

Notons que [Bey92] propose une modélisation des distorsions radiales du second ordre :

$$\begin{cases} \Delta_u &= (u - u_0)(k_1 r^2 + k_2 r^4) \\ \Delta_v &= (v - v_0)(k_1 r^2 + k_2 r^4) \end{cases}$$

Chapitre 3. Calibrage intrinsèque des caméras

Cependant dans le cas des expérimentations de [TG94], une modélisation du premier ordre semble suffisante.

Pour connaître une modélisation des autres types de distorsion, on peut se reporter à l'annexe de [Sau94].

Cependant, une modélisation physique globale n'est pas toujours appropriée. On peut alors avoir recours à une modélisation locale des distorsions, qui s'adapte davantage aux données.

3.1.2.2 Modélisation physique locale

La modélisation locale des distorsions consiste à découper l'image en régions et à déterminer un modèle pour chaque région. Dans le cas d'un modèle physique des distorsions radiales, on cherche à optimiser le paramètre k dans chaque région, les autres paramètres étant identiques pour toute l'image (remarque : on peut également chercher un modèle physique complet pour chaque région).

Jusqu'à présent, on a considéré le vecteur (Δ_u, Δ_v) comme une modélisation d'un phénomène physique : la distorsion. Cependant le recours à une modélisation locale montre les faiblesses d'une approche qui cherche à reproduire le processus exact et complet de formation d'image. En effet, le découpage en régions est fait de manière à pouvoir exploiter aisément les résultats (généralement découpage en carreaux) et ne peut donc pas reproduire des défauts qui bien sûr n'obéissent pas à des lois aussi pratiques. C'est pourquoi on s'est orienté vers des modélisations mathématiques qui elles, ne cherchent pas à modéliser explicitement le phénomène physique responsable des distorsions.

Modélisation mathématique locale

L'approche mathématique n'a plus pour objectif de reproduire le plus fidèlement possible le processus de formation des images. Elle consiste à considérer que (Δ_u, Δ_v) est une correction locale qui permet de maintenir la notion de projection perspective selon laquelle une droite est transformée en une droite. Ainsi, le pixel réel (u', v') est d'abord corrigé en pixel idéal (u, v) avant d'être exploité par la projection perspective inverse (ou rétroprojection) :

$$\begin{cases} u &= u' - \Delta_u &= f(u') \\ v &= v' - \Delta_v &= g(v') \end{cases}$$

Selon cette approche, les corrections (en l'occurrence les fonctions f et g) permettent de maintenir l'intégrité de la modélisation à sténopé. Ces fonctions sont généralement déterminées en suivant deux étapes :

- **Choix des points idéaux correspondants aux points d'intersection d'une grille régulière.**

On observe une grille dont les points d'intersection vont servir à délimiter les régions auxquelles on appliquera une correction locale. Les images de ces points d'intersection devraient constituer eux aussi une grille régulière si l'hypothèse de projection perspective était respectée. On choisit donc des points images idéaux qui vérifient cette hypothèse.

Dans la plupart des travaux, les points idéaux sont choisis simplement en appliquant le modèle à sténopé retenu (déterminé au préalable) aux points 3D de la grille observée.

[BMB94] choisit de les positionner arbitrairement en $(0, 0), (0, 1) \dots (i, j) \dots (n, m)$ dans un repère appelé repère virtuel avant de les reprojeter dans un repère proche de celui d'origine. Pour cela, P. Brand définit une transformation projective entre ces deux repères 2D, grâce à quatre points choisis comme base et pour lesquels une correction nulle est imposée a priori. Cette manière de procéder évite donc la détermination du modèle à sténopé. Elle permet également de recalculer les corrections indépendamment de la position de la mire (à condition bien sûr de choisir toujours les quatre mêmes points du repère initial), et donc d'effectuer des tests significatifs.

Quelle que soit la méthode employée, l'image est partitionnée en carreaux. Les sommets de ces carreaux sont connus dans l'image, à la fois par leurs coordonnées réelles et par leurs coordonnées idéales (ou corrigées).

– **Détermination des fonctions f et g dans chaque carreau.**

Différentes fonctions ont été utilisées dans la littérature.

[FT87] applique une correction sur les carreaux qui consiste en une interpolation bilinéaire entre les quatre sommets corrigés. Dans le $i^{\text{ème}}$ carreau, la correction s'effectue de la manière suivante :

$$\begin{cases} u &= a_i + b_i u' + c_i v' + d_i u' v' \\ v &= e_i + f_i u' + g_i v' + h_i u' v' \end{cases}$$

Il y a donc 8 paramètres à déterminer pour chaque carreau. Ces paramètres sont aisément identifiables grâce à la connaissance des coordonnées réelles et des coordonnées corrigées des 4 coins de chaque carreau. Il suffit de résoudre deux systèmes linéaires de 4 équations et 4 inconnues.

Outre l'interpolation bilinéaire, [Peu93] suggère d'autres méthodes d'interpolation entre les 4 coins de chaque carreau :

- coordonnées projectives : utilisation des birapports [LRD94]
- splines bicubiques : les valeurs corrigées u et v sont chacune la troisième coordonnée d'une surface paramétrique 2D de continuité C_1 ; ainsi les coordonnées du point milieu sont données par l'évaluation des équations de surface pour les valeurs de coordonnées locales $(0.5, 0.5)$.

Chapitre 3. Calibrage intrinsèque des caméras

Les tests de précision ne sont effectués que pour les points milieux des carreaux ayant servi à la correction. Les meilleurs résultats sont obtenus avec l'interpolation par splines bicubiques.

3.1.2.3 Précision et conclusion

[TG94] effectue un calibrage de caméra avec une modélisation des distorsions radiales. Les paramètres sont déterminés pour 12 séries de données correspondant à des positions différentes de la mire. Une variation d'environ 2% est constatée. On n'a cependant aucune idée de la précision absolue ainsi obtenue.

P. Brand utilise l'interpolation bilinéaire des quatre points de calibrage qui entourent le point considéré [BMB94]. Il annonce que sa méthode est stable, dans le temps et dans l'espace, avec une variance de 0.02 pixel et une différence maximale de 0.08 pixel, et ce malgré une certaine instabilité constatée lors de l'acquisition des images. Ces résultats sont très bons.

Cependant, la majeure partie des tests (à savoir 5) ont été effectués avec une mire placée dans la même position. 2 autres tests sont pris en compte avec la mire placée dans des positions différentes.

Il n'est cependant pas précisé dans quelle mesure la mire a été déplacée. On peut penser que la position de la mire influence les corrections d'images. Par exemple, si la mire est placée à des distances différentes de la caméra, la netteté de l'image ne peut pas être la même dans les 2 cas. La détection des points de calibrage s'en trouve altérée (de manière plus ou moins sensible selon l'algorithme utilisé). Les corrections sont donc différentes.

Les plus mauvais résultats sont d'ailleurs obtenus pour l'un des tests utilisant la mire dans une position différente.

Avec la même technique, B. Peuchot annonce une précision de 0.01 pixel pour un matériel sans doute bien choisi. Dans cette expérimentation, pour laquelle les images sont peu distordues, la méthode d'interpolation utilisée à l'intérieur de chaque carreau n'a que peu d'influence. En revanche, des tests sur des images radiographiques ont montré la supériorité de l'interpolation par splines bicubiques.

On voit que les modèles qui ne cherchent pas à déterminer explicitement les paramètres de la caméra fournissent de bons résultats. Ces modèles sont cependant encore attachés à un modèle physique du système d'acquisition d'images, dans le sens où ils tentent de reconstruire un modèle à sténopé idéal, faisant entre autre l'hypothèse d'un point focal unique.

Un autre type d'approche, purement mathématique, permet de s'affranchir de cette contrainte. Il s'agit des méthodes multi-plans.

3.1.3 Méthodes multi-plans

Nous allons uniquement étudier les méthodes bi-plans, dont les méthodes multi-plans sont une généralisation.

Avec les méthodes que nous avons étudiées jusqu'à présent et qui s'appuient sur un modèle à sténopé, on détermine d'abord la transformation qui, connaissant les coordonnées 3D d'un point, permet de calculer les coordonnées 2D de sa projection p dans l'image. On en déduit aisément la transformation inverse, la rétroprojection qui à p , associe la demi-droite de l'espace ayant pour origine le point focal et passant par p .

Dans le cas des méthodes multi-plans, on définit d'abord un modèle de rétroprojection. L'inversion de ce modèle pour obtenir un modèle de projection, n'est pas direct. Aussi allons-nous étudier successivement ces deux phases.

3.1.3.1 La rétroprojection

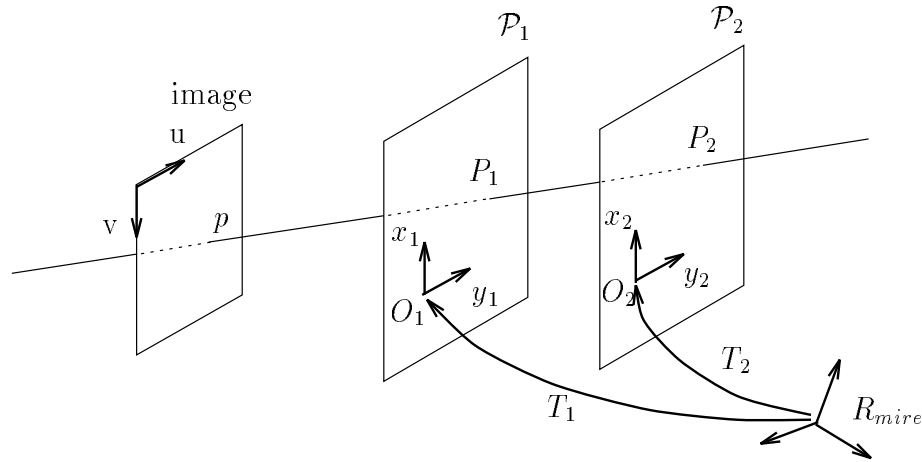


FIG. 3.2 - Méthode des 2 plans

Le principe consiste à reconstruire pour chaque pixel de l'image, deux points P_1 et P_2 situés respectivement dans deux plans de calibrage $\mathcal{P}_1$ et $\mathcal{P}_2$. Les points P_1 et P_2 définissent la droite de rétroprojection associée au pixel considéré. On voit donc que les lignes de vue ne sont plus forcément concourantes. Notons que dans le cas d'un modèle utilisant n plans, la droite de rétroprojection peut être définie comme la droite minimisant aux moindres carrés la distance aux points $P_1 \cdots P_i \cdots P_n$.

On décompose la détermination de chaque point P_i en deux phases.

La première phase consiste à associer à un pixel (u, v) de l'image, les coordonnées (x_i, y_i) du point correspondant P_i dans le plan $\mathcal{P}_i$. Les coordonnées (x_i, y_i) sont exprimées dans R_i , un repère associé au plan $\mathcal{P}_i$ (les coordonnées de (P_i) sont en fait $(x_i, y_i, 0)$).

Chapitre 3. Calibrage intrinsèque des caméras

La deuxième phase consiste à déterminer les coordonnées (X_i, Y_i, Z_i) du point P_i dans un repère 3D absolu, R_{mire} associé à la mire de calibrage.

Exprimé à l'aide de coordonnées homogènes, on a donc :

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = T_i \cdot \begin{pmatrix} x_i \\ y_i \\ 0 \\ 1 \end{pmatrix}$$

Les matrices T_i sont généralement connues par construction de la mire, le repère absolu R_{mire} étant choisi dans la position initiale de la mire. On peut se reporter à [CLSB92] pour une description des matrices de passage T_i dans le cas d'une mire de calibrage à 3 degrés de liberté (2 rotations et une translation).

Dans le cas où la translation subie par le plan de calibrage est perpendiculaire au plan lui-même, et de longueur relative Z , les matrices de passage peuvent s'exprimer simplement en choisissant comme repère absolu le repère associé au plan de calibrage dans la première position. Alors :

$$T_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{et} \quad T_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ainsi les points de chaque plan peuvent être exprimés dans un repère commun.

Revenons à la première phase. Des formules d'interpolation indépendantes sont calculées pour chaque plan, grâce à un ensemble de points de calibrage collectés sur le plan courant. Ces formules permettent d'associer à un pixel (u, v) de l'image, les coordonnées (x_i, y_i) du point correspondant P_i appartenant au plan $\mathcal{P}_i$.

La littérature recense divers types d'interpolation. [MBK81] présente les trois premiers :

- **Interpolation linéaire** : Soit $(u \ v)^t$ un pixel de l'image. Soit $P_i = (x_i \ y_i \ 0)^t$ le point du plan $\mathcal{P}_i$ associé au pixel $(u \ v)^t$. On a :

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = A_i \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad \text{c'est-à-dire} \quad P_i = A_i \cdot L$$

3 points sont nécessaires pour déterminer les paramètres, c'est-à-dire les 6 coefficients de la matrice A_i (de taille 2x3). Dans la pratique, on collecte davantage de points et l'on utilise la technique de la pseudo-inverse pour minimiser les erreurs aux moindres carrés.

- **Interpolation quadratique** : Cette fois-ci, l'approximation est du second ordre dans chaque plan : $P_i = A_i \cdot Q$ avec $Q = (u^2 \ v^2 \ uv \ u \ v \ 1)^t$ et A_i de taille 2x6.

6 points sont nécessaires à la détermination des paramètres. Mais comme dans le cas linéaire, on en collecte davantage et l'on minimise les erreurs aux moindres carrés.

Il est facile d'étendre cette manière de procéder à des approximations d'ordre plus élevé. Ainsi, pour une interpolation cubique, on aurait 10 paramètres à déterminer.

- **Interpolation par splines linéaires** : Dans chaque plan, les trois points les plus proches définissent un triangle contenant le point à interpoler. Les sommets des triangles sont utilisés pour calculer une approximation linéaire locale. On a alors : $P_i = A_{ij}.L$ où $i = 1, 2$ indice les plans et $j = 1 \dots N$ indice les triangles.

La matrice A_{ij} est la solution exacte du système d'équations constitué par les coordonnées dans le plan i et dans l'image, des trois sommets du triangle j .

Notons que l'on pourrait utiliser des carreaux (donc 4 points) au lieu de triangles et procéder de la même manière que pour la correction des distorsions avec une interpolation bilinéaire locale (cf 3.1.2.2).

La précision est d'autant meilleure que le nombre de points de calibrage, donc le nombre de triangles ou carreaux, est important. La contre-partie de cette amélioration est que le temps de calcul et la place de stockage sont également plus importants.

- **Interpolation par splines bicubiques pseudo plaque mince** : Cette méthode a été introduite par [CLSB92] et apporte une formulation globale de l'interpolation par des splines. La méthode des splines linéaires exposée par [MBK81] impose la recherche du triangle contenant le point considéré. Elle est donc coûteuse en temps de calcul. Ici, le découpage ne se fait plus selon des triangles aléatoires, mais selon des carrés réguliers, bien que les points de calibrage puissent être répartis irrégulièrement. On utilise 4 fonctions d'interpolation pour déterminer les coordonnées (x_1, y_1) dans le plan $\mathcal{P}_1$ et (x_2, y_2) dans le plan $\mathcal{P}_2$.

$$\text{On a :} \quad \begin{array}{ll} x_1 &= f_1(u, v) & x_2 &= f_2(u, v) \\ y_1 &= g_1(u, v) & y_2 &= g_2(u, v) \end{array}$$

Ces fonctions sont de la forme :

$$f(u, v) = \sum_{m=-3}^{n_u-1} \sum_{l=-3}^{n_v-1} \alpha_{m,l} B_m(u) B_l(v)$$

B_m et B_l sont les composantes de la base de splines [BFK84] et les coefficients $\alpha_{m,l}$ sont déterminés en minimisant une fonctionnelle qui tend à la fois à lisser la fonction cherchée f , et à interpoler les données. Les coefficients permettent donc de rendre compte des distorsions locales.

Chapitre 3. Calibrage intrinsèque des caméras

Notons que si l'erreur d'interpolation est inférieure à l'erreur de mesure des données de calibrage, on ne perd rien en précision.

Les tests effectués par [MBK81] ont montré la supériorité des méthodes bi-plans sur un simple modèle à sténopé.

Dans les tests effectués par [CLSB92], la meilleure précision est obtenue pour l'interpolation par splines bicubiques pseudo plaque mince, puis pour les splines linéaires et enfin la méthode quadratique, les autres types de calibrage étant largement supplantés. Les méthodes locales (splines linéaires et splines bicubiques) sont néanmoins relativement coûteuses en temps de calcul, à la fois lors de la phase de calibrage et lors des calculs de rétroprojection. Le modèle quadratique, bien que légèrement moins précis, semble suffisant dans de nombreux cas.

3.1.3.2 La projection

Le modèle à sténopé est facilement inversible, puisqu'il est totalement linéaire. Ce n'est plus le cas avec les méthodes multi-plans.

Du fait que la précision exigée pour la projection est généralement peu importante, on peut utiliser en parallèle un modèle à sténopé. C'est le cas du système décrit ici, puisque la projection sert uniquement à prédire une zone de l'image où chercher des indices intéressants.

[GTK88] a cependant élaboré une méthode plus précise (0.34 pixels). Le principe consiste à construire des modèles à sténopé locaux.

Pour cela, on commence par déterminer un pseudo point focal en cherchant le point 3D qui minimise aux moindres carrés la somme des distances à un certain nombre de droites de rétroprojection. Les droites de rétroprojection en question sont bien sûr choisies dans la région locale d'intérêt. Une fois le pseudo point focal estimé, la détermination des autres paramètres consiste simplement à résoudre un système linéaire.

Soit P un point 3D dont on veut déterminer la projection dans l'image. La technique complète consiste à utiliser un modèle à sténopé global pour déterminer une projection grossière (u_{gross}, v_{gross}) de P , puis à choisir le modèle à sténopé M approprié en fonction de (u_{gross}, v_{gross}) , et enfin à projeter P grâce à M .

3.1.4 Calibrage à focale variable

Pour représenter un objectif à focale variable, Lavest, Rives et Dhome [LRD92] ont essayé d'utiliser un modèle à sténopé dont les paramètres liés physiquement au changement de focale, f et T_z , seraient variables. f désigne la longueur focale et T_z , la translation selon l'axe optique. Mais ils ont décelé des incohérences avec ce modèle puisque leurs expérimentations ont montré que les variations de f et de T_z ne se compensent pas.

Ils se sont donc orientés vers un modèle de lentille plus complexe, à savoir un modèle épais. Ils ont fait l'approximation de Gauss qui consiste à considérer que les

rayons lumineux sont paraxiaux, c'est-à-dire qu'ils sont proches de l'axe optique. Pour un tel modèle de lentille, le centre optique défini dans un modèle à sténopé est remplacé par les points nodaux avant et arrière.

Le zoom Angénieux utilisé a été simulé à partir des plans du brevet. Les résultats de simulation sont en accord avec les résultats d'une série de calibrages effectués pour différentes valeurs de focales. Cette bonne adéquation provient vraisemblablement de l'excellente qualité de l'objectif utilisé qui effectue un changement de focale par une translation pure des deux blocs de lentilles qui le constituent. D'ailleurs, d'autres expérimentations sur des objectifs de moins bonne qualité ont montré que le point principal³ varie lorsque l'on modifie la focale [WW90].

Tarabanis, Tsai et Goodman [TTG92] décrivent une méthode pour déterminer les trois paramètres optiques effectifs du modèle de lentille épaisse (d_{eff} , la distance entre le point nodal arrière et le plan image, f_{eff} , la longueur focale effective et a_{eff} , le diamètre de la pupille d'entrée) en fonction des trois paramètres contrôlables de l'objectif étudié (z_c , le zoom, map_c , le focus ou la mise au point et a_c , l'ouverture). Les mesures sont faites pour chaque triplet (z_c, map_c, a_c) désiré. Par la suite, l'objectif ne peut être utilisé que pour ces triplets calibrés. Cette méthode permet cependant de choisir les paramètres commandables pour obtenir des paramètres effectifs proches de ceux désirés.

Toutes les manipulations décrites dans [TTG92] sont très complexes et ne se justifient pas si l'on veut simplement connaître un modèle du système optique pour un ensemble discret de n-uplets de paramètres contrôlables. Dans ce cas-là, il suffit de calibrer indépendamment la caméra pour chacun de ces n-uplets. C'est ce qui est fait dans [RA94] avec un modèle à sténopé.

Encisco, Viéville et Faugeras [EVF94] vont plus loin puisqu'ils proposent d'ajuster les paramètres intrinsèques d'un système optique au fur et à mesure qu'il est modifié, grâce au suivi dans les images de plusieurs points qui restent fixes dans la scène observée. Ils supposent que le changement de mise au point ou de focale revient à effectuer une transformation affine dans le plan image. Cette transformation, notée A_{aff} , permettrait de calculer les nouveaux paramètres intrinsèques à partir des anciens. Ils font l'hypothèse que les paramètres k_u, k_v (taille d'un pixel) et θ (angle entre les deux axes du plan image, très proche de $\frac{\pi}{2}$) sont invariants. Grâce à ces hypothèses, A_{aff} est de la forme :

$$A_{aff} = \begin{pmatrix} C_0 & 0 & a_0 \\ 0 & C_0 & b_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Pour déterminer les trois inconnues a_0, b_0 et C_0 , il suffit de suivre deux points.

Cette méthode est bien adaptée à la vision active puisqu'elle est très rapide et qu'elle est performante si l'on se restreint à une région localisée de l'image. Ce-

3. Le point principal est le point 2D du plan image par lequel passe l'axe optique de la caméra. Dans le modèle à sténopé de la figure 3.1, ses coordonnées exprimées en pixels sont (u_0, v_0) .

Chapitre 3. Calibrage intrinsèque des caméras

pendant, elle ne permet pas d'effectuer des mesures précises ([EVF94] annonce une précision de 1.5 pixels dans le cas d'un changement de mise au point et 5 pixels pour un changement de focale). De plus elle n'est valable que de manière locale.

Willson [Wil94] effectue le calibrage de son système optique pour un ensemble discret de couples (focale, mise au point). Ces calibrages sont effectués selon la méthode de Tsai [Tsa87], c'est-à-dire que les paramètres intrinsèques considérés sont : s , le facteur d'échelle entre les axes x et y , f , la longueur focale, u_0 et v_0 , les coordonnées du point principal, et k_1 , le coefficient de distorsion radiale. Il cherche ensuite à construire un modèle continu du système optique. Pour cela, il approxime chacun des paramètres par un polynôme dont le degré est choisi expérimentalement en fonction de la précision que l'on attend et de la manière dont varient les données. s et les paramètres extrinsèques R_x , R_y , R_z , T_x et T_y sont considérés constants. Les paramètres T_z , f , u_0 et v_0 sont modélisés par un polynôme de degré 5 et k_1 par un polynôme de degré 2.

Les polynômes ne sont pas déterminés indépendamment en une seule étape, mais sont ajustés successivement. Pour chaque couple (focale, mise au point), les paramètres sont estimés à partir des données de calibrage. Puis, les valeurs du premier paramètre sont approximées par un polynôme. Pour chaque couple (focale, mise au point), les paramètres restants sont réestimés à partir des données de calibrage. Les valeurs du deuxième paramètre sont alors approximées par un polynôme. Ce processus est répété jusqu'à ce que tous les paramètres aient été modélisés.

L'erreur moyenne annoncée est de 0.11 pixels. Remarquons qu'elle a été mesurée sur les données de calibrage. Elle représente donc l'erreur d'approximation et non pas l'erreur réelle. On peut cependant penser que cette méthode est assez précise, étant donné le degré des polynômes d'approximation. Elle présente au moins l'intérêt de fournir un modèle même pour des couples (focale, mise au point) pour lesquels la caméra n'a pas été explicitement calibrée.

Dans le même ordre d'idées, nous avons étendu la méthode des deux plans avec une modélisation bicubique, au calibrage d'un objectif à focale variable. On dispose de deux plans de calibrage repérés dans un repère absolu par leur matrice de passage T_i . On associe à un pixel (u, v) de l'image et à une focale f , les coordonnées (x_i, y_i) du point correspondant dans le plan $\mathcal{P}_i$. Les fonctions d'interpolation dépendent donc de trois paramètres au lieu de seulement deux dans le cas du calibrage à focale fixe.

$$\text{On a :} \quad \begin{array}{lll} x_1 & = & f_1(u, v, f) \\ y_1 & = & g_1(u, v, f) \end{array} \quad \begin{array}{lll} x_2 & = & f_2(u, v, f) \\ y_2 & = & g_2(u, v, f) \end{array}$$

Ces fonctions sont de la forme :

$$f(u, v, f) = \sum_{m=-3}^{n_u-1} \sum_{l=-3}^{n_v-1} \sum_{k=-3}^{n_f-1} \alpha_{m,l,k} B_m(u) B_l(v) B_k(f)$$

Cette méthode ne cherche pas à modéliser le phénomène physique de formation des images. Elle est donc moins contrainte que toutes les autres approches. Ceci

lui permet de s'ajuster aux données de calibrage ce qui est tout-à-fait approprié ici puisque les variations des fonctions impliquées sont relativement lisses.

Cette méthode étant purement mathématique, elle est applicable non seulement à la variation de la focale, mais de manière plus général, à la variation de tout paramètre d'une caméra. La mise au point ou l'ouverture en sont deux exemples. On peut même envisager la variation simultanée de plusieurs paramètres, les contraintes étant la quantité importante de données de calibrage nécessaires et la complexité croissante de la modélisation de chaque plan.

Nous n'avons effectué des tests que pour une variation de la focale. Les résultats que nous avons obtenus sur des données différentes de celles utilisées pour le calibrage sont présentés lors de la troisième expérience du paragraphe 3.2.3.3. L'erreur moyenne obtenue est de 1.2 mm et l'erreur maximale de 2.8 mm pour un champ d'environ $800 \times 700 \times 700 \text{ mm}^3$ et des focales telles que $\alpha_u = \frac{f}{k_u}$ varie de 2400 à 5000.

3.1.5 Conclusion - Méthodes retenues

Dans le cadre d'un calibrage à focale fixe, nous choisissons d'expérimenter la méthode des deux plans avec une modélisation polynômiale cubique qui semble offrir un bon compromis de précision et de rapidité de calcul.

Dans le cadre d'un calibrage à focale variable, nous avons proposé une nouvelle méthode de calibrage qui est une extension de la méthode des deux plans avec une modélisation par splines bicubiques utilisée pour un calibrage de caméra dont les paramètres sont fixes.

Dans la deuxième partie de ce chapitre, nous allons décrire les expérimentations que nous avons menées pour évaluer la précision des méthodes que nous avons retenues.

3.2 Mise en oeuvre et résultats

Nous allons présenter les spécificités de la mire de calibrage que nous avons utilisée et les divers tests effectués : l'étude de la répétabilité de l'acquisition d'images permettra de savoir dans quelle mesure les imprécisions sont dues à un calibrage médiocre ou à des défauts inhérents au matériel employé ; nous procéderons alors à des tests de précision concernant le calibrage de caméras proprement dit.

3.2.1 Présentation du matériel

3.2.1.1 Le système d'acquisition d'images

La caméra utilisée pour les tests est un module de caméscope standard SONY. Elle est connectée à une carte d'acquisition vidéo Rasterops installée sur une station de travail DEC DS3100.

3.2.1.2 La mire de calibrage

La mire utilisée mesure $765 \times 765 \text{ mm}^2$. Elle est montée sur un rail permettant de la translater perpendiculairement à son plan, selon une amplitude connue. Elle est constituée d'une plaque de plexiglas blanc recouverte d'un film noir. Elle a été usinée pour supprimer le film en certains endroits (138 disques et 3 droites).

Lorsqu'elle est éclairée par derrière et qu'elle est observée par une caméra, la mire présente un fort contraste entre le fond noir et les indices de calibrage blancs. Contrairement aux mires illuminées par devant, cette technique offre l'avantage de s'affranchir de l'hypothèse lambertienne selon laquelle une surface rayonne de façon diffuse une fraction de la lumière reçue.

Les cercles de 11 mm de diamètre sont quasiment tous répartis selon un quadrillage régulier de 62 mm . Ce sont leurs centres qui vont constituer nos données de calibrage précises. La position des centres est connue avec une précision de 0.1 mm .

Les trois droites s'intersectent en trois points dont les coordonnées sont connues avec une précision de 0.1 mm . Elles servent de points de repère et permettent d'identifier les trous observés par la caméra.

3.2.2 Répétabilité de l'acquisition d'images

Le système d'acquisition d'images est constitué d'une multitude de composants électroniques très sensibles. Une utilisation prolongée de la caméra peut donc se traduire par un comportement non uniforme. D'autre part, des études [BC92, HK92] ont montré la présence de bruit inhérent à la technologie des caméras CCD (phénomène d'ionisation de cellules).

L'objectif des expériences qui suivent a été de déterminer la répétabilité du système expérimental.

Première expérience : acquisition d'une série de 10 images consécutives.

La mire et le système d'acquisition d'images sont immobiles. On acquiert 10 images dans lesquelles on détecte la position des points de calibrage en calculant le barycentre des tâches. Soient $u_i(k)$ et $v_i(k)$, les coordonnées du point numéro k dans la $i^{\text{ème}}$ image.

Soient $u_{\text{moy}}(k)$ et $v_{\text{moy}}(k)$, les coordonnées moyennes du $k^{\text{ème}}$ point de la mire :

$$P_{\text{moy}}(k) = \begin{pmatrix} u_{\text{moy}}(k) \\ v_{\text{moy}}(k) \end{pmatrix} = \frac{1}{N_{\text{images}}} \sum_{i=1}^{N_{\text{images}}} \begin{pmatrix} u_i(k) \\ v_i(k) \end{pmatrix}$$

On peut alors calculer les distances maximale et moyenne au point moyen, ainsi que l'écart-type :

$$\begin{aligned} \text{max}_k &= \text{Max}_{i=1..N_{\text{images}}} d(P_i(k), P_{\text{moy}}(k)) \\ \text{moy}_k &= \frac{1}{N_{\text{images}}} \sum_{i=1}^{N_{\text{images}}} d(P_i(k), P_{\text{moy}}(k)) \end{aligned}$$

$$\sigma_k = \left(\frac{1}{N_{images}} \sum_{i=1}^{N_{images}} d^2(P_i(k), P_{moy}(k)) \right)^{\frac{1}{2}}$$

$$\text{avec } d(P_i(k), P_{moy}(k)) = \sqrt{(u_i(k) - u_{moy}(k))^2 + (v_i(k) - v_{moy}(k))^2}$$

On ne retient que les valeurs moyenne et maximale de ces expérimentations (distances moyenne et maximale au point moyen et écart-type pour chaque point). Ces résultats sont bien sûr exprimés en pixels.

	Moyenne ($f(k) = moy_k$)	Maximum ($f(k) = max_k$)	Ecart type ($f(k) = \sigma_k$)
Valeur moyenne ($\frac{1}{N_{points}} \sum_{k=1}^{N_{points}} f(k)$)	0.03	0.06	0.01
Valeur maximale ($Max_{k=1}^{N_{points}} f(k)$)	0.06	0.13	0.03

Deuxième expérience : acquisition de 14 séries de 10 images espacée de 30 minutes.

Chaque série de 10 images est acquise consécutivement. L'acquisition de 2 séries est séparée de 30 minutes.

On effectue sur chaque série la même expérience que précédemment. Les résultats sont tout-à-fait similaires dans tous les cas (en moyenne, erreur maximale de l'ordre de 0.06 pixels, erreur moyenne de l'ordre de 0.03 pixels, écart-type de l'ordre de 0.01 pixels).

Pour chaque série i , on conserve l'ensemble des points moyens $P_{moyi}(k)$ pour $k = 1..N_{points}$. C'est sur cet ensemble de points que nous allons étudier la stabilité dans le temps.

On définit les écarts maximal et moyen entre 2 séries de mesures i et j par :

$$max_{ij} = Max_{k=1..N_{points}} d(P_{moyi}(k), P_{moyj}(k))$$

$$moy_{ij} = \frac{1}{N_{points}} \sum_{k=1}^{N_{points}} d(P_{moyi}(k), P_{moyj}(k))$$

$$\text{avec } d(P_{moyi}(k), P_{moyj}(k)) = \sqrt{((u_{moyi}(k) - u_{moyj}(k))^2 + (v_{moyi}(k) - v_{moyj}(k))^2)}$$

Les courbes de la figure 3.3 montrent l'évolution dans le temps des écarts entre 2 images moyennes. Le premier schéma concerne l'écart entre 2 images acquises successivement (c'est-à-dire dont les acquisitions sont séparées de 30 mn). Le deuxième schéma montre l'écart entre l'image moyenne acquise au temps $t = k * 30 \text{ mn}$ ($k = \{1, 2, \dots, 13\}$) et l'image moyenne acquise au temps $t = 0$.

Ces résultats montrent que la stabilité dans le temps n'est pas parfaite. On constate une légère dérive (en moyenne 0.1 pixel en 6h30) qui risque de légèrement

Chapitre 3. Calibrage intrinsèque des caméras

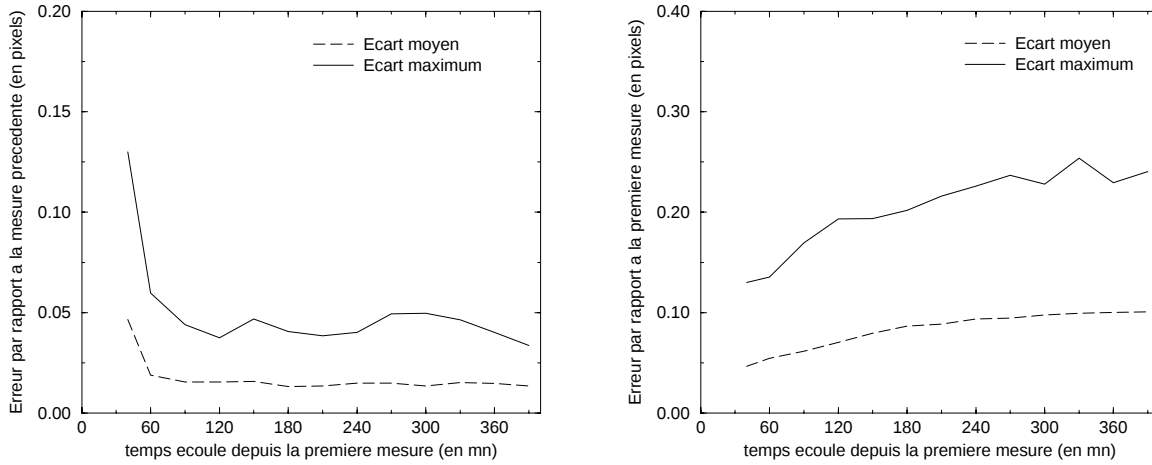


FIG. 3.3 - *Stabilité de l'acquisition d'images*

altérer la précision de la surveillance. Ces expériences ont été menées à plusieurs reprises. L'allure et l'amplitude des courbes sont toujours similaires à celles de la figure 3.3.

Troisième expérience : On cherche à tester la répétabilité du repositionnement du zoom. Pour cela, on choisit une valeur focale $f = f_0$. On commande le module de camescope pour l'amener à cette valeur et on acquiert une image. On modifie la valeur de la focale et on l'amène à nouveau à $f = f_0$. On effectue cette manoeuvre 10 fois et l'on mesure la stabilité du repositionnement de la même manière que lors de la première expérience.

Les résultats obtenus pour la valeurs $f_0 = 565$ pas codeurs sont décrits par les trois tableaux suivants.

Ce premier tableau montre les résultats obtenus en amenant le zoom à la valeur $f_0 = 565$ toujours à partir de la même valeur $f_1 = 465$.

	Moyenne ($f(k) = moy_k$)	Maximum ($f(k) = max_k$)	Ecart type ($f(k) = \sigma_k$)
Valeur moyenne ($\frac{1}{N_{points}} \sum_{k=1}^{N_{points}} f(k)$)	0.11	0.16	0.02
Valeur maximale ($Max_{k=1}^{N_{points}} f(k)$)	0.21	0.29	0.05

Ce deuxième tableau montre les résultats obtenus en amenant le zoom à la valeur $f_0 = 565$ toujours à partir de la même valeur $f_1 = 649$.

	Moyenne ($f(k) = moy_k$)	Maximum ($f(k) = max_k$)	Ecart type ($f(k) = \sigma_k$)
Valeur moyenne ($\frac{1}{N_{points}} \sum_{k=1}^{N_{points}} f(k)$)	0.12	0.21	0.05
Valeur maximale ($Max_{k=1}^{N_{points}} f(k)$)	0.21	0.40	0.10

Enfin le dernier tableau montre les résultats obtenus en amenant le zoom à la valeur $f_0 = 565$ à partir de valeurs f_1 variables (entre 460 et 670).

	Moyenne ($f(k) = moy_k$)	Maximum ($f(k) = max_k$)	Ecart type ($f(k) = \sigma_k$)
Valeur moyenne ($\frac{1}{N_{points}} \sum_{k=1}^{N_{points}} f(k)$)	0.21	0.37	0.06
Valeur maximale ($Max_{k=1}^{N_{points}} f(k)$)	0.40	0.67	0.10

Nous avons également effectué ces mêmes mesures pour d'autres valeurs de focales et avons obtenu des résultats du même ordre de grandeur. On remarque bien sûr que la répétabilité est meilleure lorsque l'on vient d'une position identique. Il sera donc intéressant d'utiliser le comescope de cette manière.

On peut se reporter à [BMB94] pour une étude de l'influence de l'ouverture de l'iris et du réglage de la netteté sur l'instabilité des distorsions.

3.2.3 Précision du calibrage

3.2.3.1 Méthode d'évaluation de la précision

Pour évaluer la précision du calibrage de caméra, on utilise deux plans de calibrage ($\mathcal{P}_2$ et $\mathcal{P}_4$) et trois plans de tests ($\mathcal{P}_1$, $\mathcal{P}_3$ et $\mathcal{P}_5$).

Rappelons que le principe de la méthode des deux plans consiste à associer à chaque pixel (u, v) , un point 3D situé dans chacun des deux plans de calibrage. Ces deux points déterminent $\mathcal{D}$, la droite de rétroprojection de (u, v) .

Pour effectuer les tests de précision :

- On place la mire dans une troisième position, qui peut être avant (plan $\mathcal{P}_1$), après (plan $\mathcal{P}_5$), ou entre (plan $\mathcal{P}_3$) les deux plans de calibrage. Cette position est repérée à la précision près du système mécanique de déplacement ou d'un localisateur externe (en l'occurrence Optotrak). Nous verrons au paragraphe suivant, la précision qu'il est possible d'atteindre selon ces deux modes de repérage. On connaît donc la position 3D théorique de chacun des trous de la

Chapitre 3. Calibrage intrinsèque des caméras

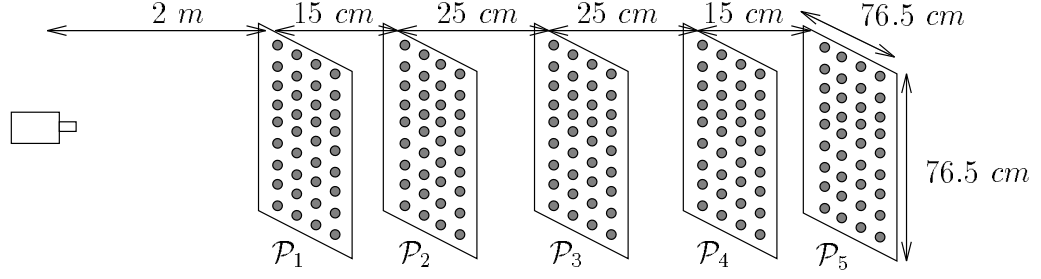


FIG. 3.4 - Dispositif expérimental pour l'évaluation de la précision du calibrage

mire dans le référentiel absolu de calibrage R_{mire} , d'où un ensemble de points $(P_k)_{R_{mire}}$.

- On saisit une image de la mire dans cette position, et on calcule précisément le centre (u_k, v_k) de chaque trou dans l'image. Pour cela, on utilise bien sûr le même algorithme que lors du calibrage proprement dit. Dans notre cas, il s'agit d'un "barycentre amélioré".
- On calcule la droite de rétroprojection associée $\mathcal{D}_k$ dans le référentiel R_{mire} .
- La distance $d(P_k, \mathcal{D}_k)$ permet alors de connaître la précision du calibrage de caméra (voir figure 3.5). Pour un calibrage idéal, la valeur d serait nulle.

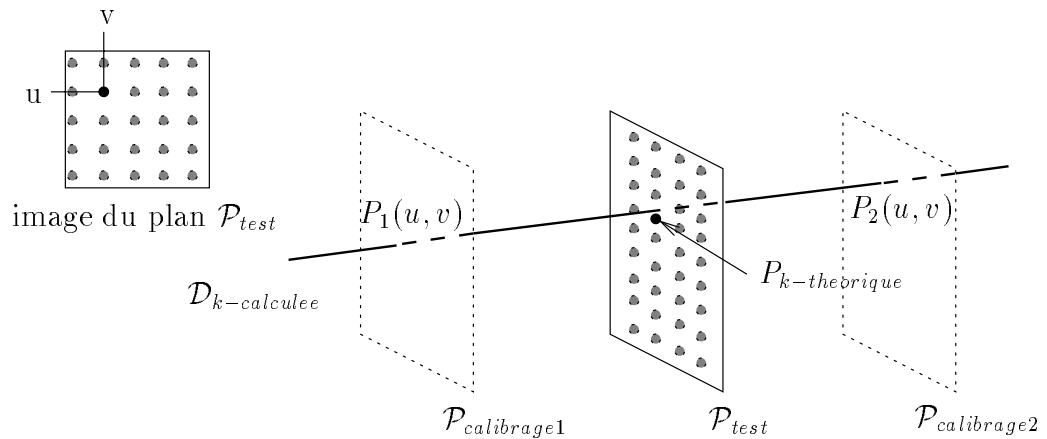


FIG. 3.5 - Méthode d'évaluation de la précision du calibrage

3.2.3.2 Erreurs engendrées par les systèmes de repérage de la position des plans de calibrage et des plans de test

Premier système de repérage : le localisateur Optotrak (voir l'annexe A)

On fixe un “rigid body” sur la mire de calibrage afin de la localiser grâce à l'optotrak.

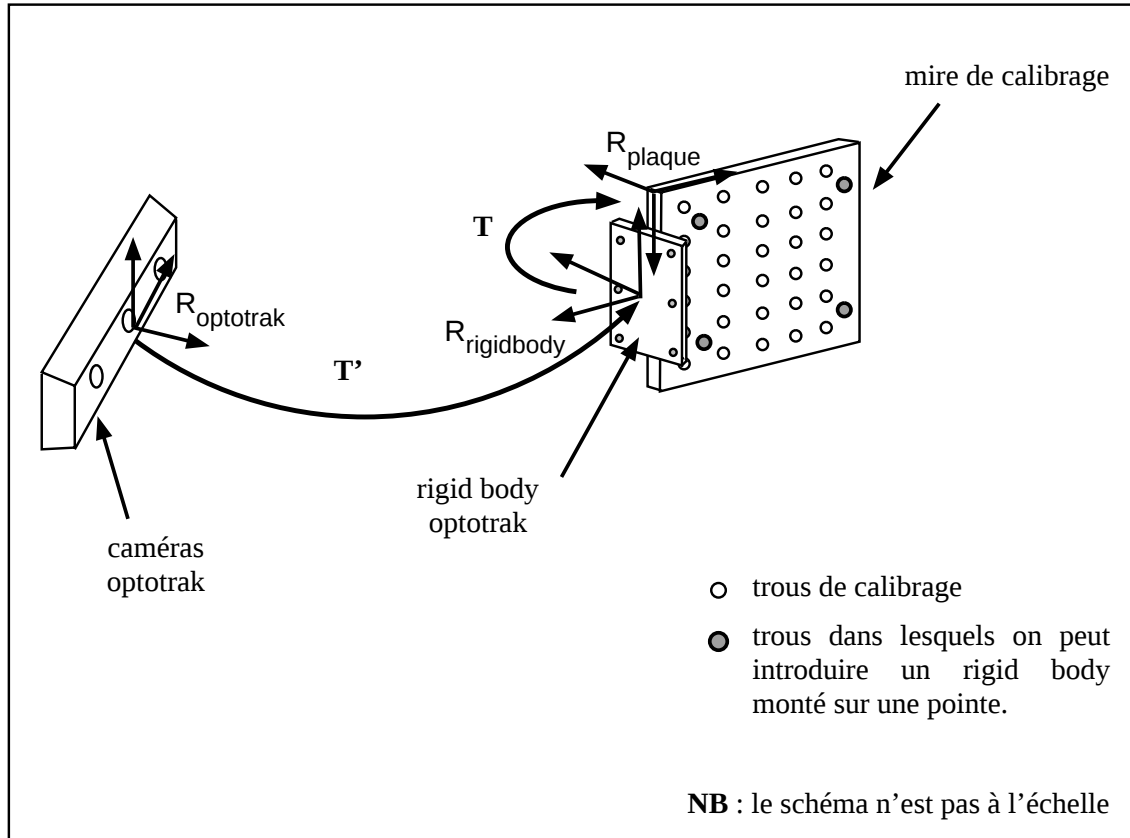


FIG. 3.6 - Repérage de la mire de calibrage par le localisateur Optotrak

La localisation se fait en deux étapes :

- La première étape consiste à déterminer la matrice de passage T entre le repère associé à la mire de calibrage, désigné par R_{plaque} sur le schéma 3.6 et le repère associé au “rigid body”, désigné par $R_{rigidbody}$. Cette matrice est invariante tant que l'on ne démonte pas le “rigid body” de la mire. La mire comporte 4 trous dans lesquels on peut introduire une pointe montée sur un autre “rigid body” (photo 3.7). Cette pointe est équipée d'un anneau qui permet de bloquer le “rigid body” contre la mire (figure 3.8).

La position de ces 4 trous est connue avec une précision de 0.1 mm dans le repère R_{plaque} .

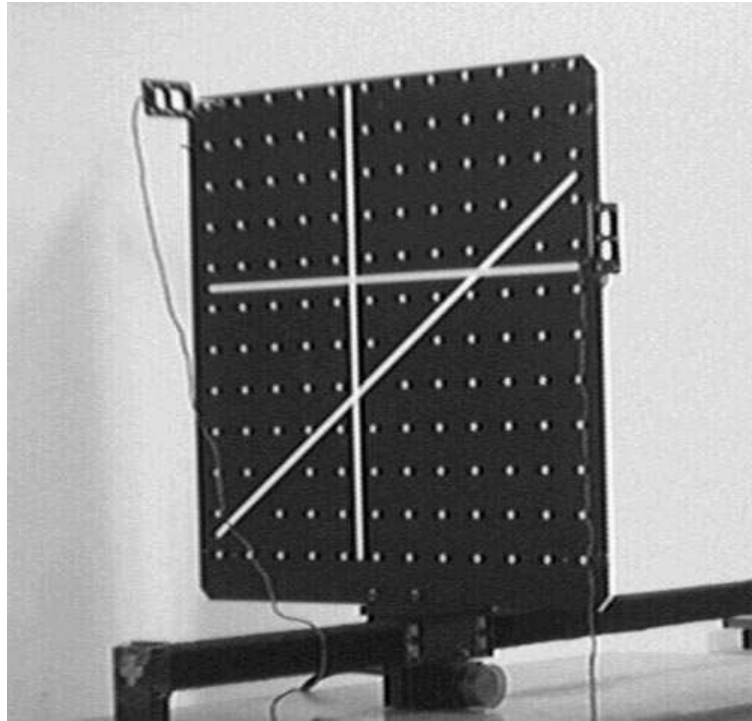


FIG. 3.7 - Mire comportant 4 trous dans lesquels on peut introduire une pointe annelée

T est déterminée de la manière suivante, selon une méthode développée par Stéphane Lavallée :

- La pointe est calibrée, c'est-à-dire que l'on connaît la position de l'extrémité dans le repère associé au rigid body qui la porte. Ceci peut être réalisé par une méthode de point fixe (voir l'annexe A).
- On introduit la pointe dans chacun des 4 trous, d'un côté de la plaque, puis de l'autre.

Optotrak calcule les coordonnées de la pointe dans le référentiel $R_{rigidbody}$. La moyenne de ces 2 points correspond au point situé au centre du trou au milieu de l'épaisseur de la plaque. On connaît donc les coordonnées

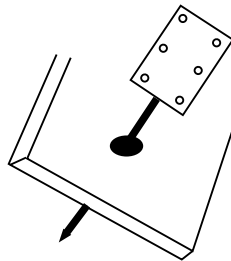


FIG. 3.8 - "Rigid body" optotrak muni d'une pointe annelée

des 4 trous dans le référentiel $R_{rigidbody}((P_i)_{R_{rigidbody}}$ pour $i = 1..4$).

Or on connaît également les coordonnées de ces 4 points dans R_{plaque} par construction de la mire : $(P_i)_{R_{plaque}}$ pour $i = 1..4$.

Par une méthode de mise en correspondance point par point [AHB87], on calcule la matrice de passage entre R_{plaque} et $R_{rigidbody}$:

$$(P)_{R_{rigidbody}} = T.(P)_{R_{plaque}}$$

- La deuxième étape consiste simplement à lire la matrice T' courante, fournie directement par le système Optotrak.

La combinaison des matrices T et T' fournit la position de la mire de calibrage dans le repère $R_{optotrak}$.

Une estimation grossière de la précision de l'Optotrak a été effectuée de la manière suivante : On acquiert la matrice de passage T_1 entre le référentiel de l'Optotrak et le référentiel du “rigid body” fixé sur la mire dans cette première position. On déplace la mire selon une translation dont l'amplitude est connue à 0.1 mm près grâce à l'unité de translation motorisée. On acquiert la matrice de passage T_2 entre le référentiel de l'Optotrak et le référentiel du “rigid body” dans cette deuxième position.

On mesure la translation entre ces deux positions :

$$\|\vec{t}\| = \sqrt{(T_1[0][3] - T_2[0][3])^2 + (T_1[1][3] - T_2[1][3])^2 + (T_1[2][3] - T_2[2][3])^2}$$

Lorsque le “rigid body” placé sur la mire est bien dans le champ de l'Optotrak, et pour des déplacements inférieurs à 600 mm , on obtient une erreur moyenne de 0.16 mm et une erreur maximale de 0.30 mm entre la translation de référence fournie par l'unité de translation motorisée et la translation fournie par l'Optotrak.

Les rotations des deux matrices de passages sont théoriquement identiques, puisqu'on a simplement translaté le plan de la mire. Cependant, principalement à cause de l'imprécision de l'Optotrak, elles ne le sont pas. Pour évaluer l'influence de cette imprécision en rotation, on procède ainsi : on applique chacune des 2 matrices de rotation à un point P ; on obtient ainsi 2 points P' et P'' ; on calcule la distance entre P' et P'' .

Le point P est choisi de manière à représenter une situation réaliste qui maximise l'erreur possible : le point appartient à la mire et il est situé le plus loin possible du repère associé au “rigid body” optotrak ; on a choisi $P = (500, 0, 765)$ (ces valeurs correspondent à la configuration de nos tests).

Sur une vingtaine de mesures effectuées pour un déplacement de 600 mm , l'erreur moyenne vaut 0.74 mm et l'erreur maximale 1.37 mm .

En revanche, si l'on est en limite de champ, les erreurs peuvent être très importantes. C'est souvent le cas lorsqu'on déplace la mire de 800 mm , ce qui correspond

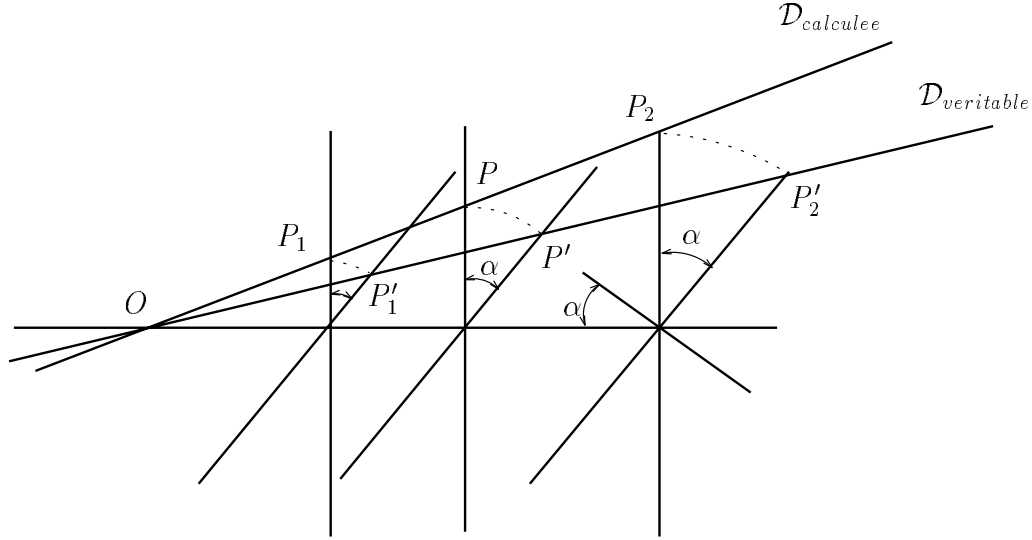


FIG. 3.10 - Les erreurs engendrées par l'angle α entre le vecteur déplacement et la normale au plan de la mire sont absorbées par le fait que le plan ne subit que des translations. Si l'on suppose que l'angle α est nul, on pense que la position d'un point de test P_{test} est P et que la droite de rétroprojection associée à l'image de P_{test} est décrite par les points P_1 et P_2 . En réalité (angle α non nul), le point de test se trouve en P' , et la véritable droite de rétroprojection est décrite par P'_1 et P'_2 . L'erreur calculée selon la méthode du paragraphe précédent portera sur les coordonnées de P , P_1 et P_2 . Il faut donc lui ajouter l'erreur e décrite par la figure précédente.

Pour simplifier les calculs, on se place dans le cas où $\vec{n}$ est dans un plan vertical, c'est-à-dire que l'on considère que la mire n'a subi qu'une rotation autour de l'axe horizontale perpendiculaire à $\vec{t}$.

D'autre part, on fait les hypothèses suivantes :

- La mire est placée à une distance d de O , le centre optique de la caméra.
- La projection de O selon $\vec{t}$ sur la mire, se trouve à peu près au milieu de la mire.

Alors on va étudier l'erreur commise sur un point P situé sur le bord supérieur de la mire. C'est sur ces points que l'erreur est la plus grande. Si l'on suppose une perpendicularité parfaite, le point P sera placé en A . On lui associera alors la droite de rétroprojection $\mathcal{D}$ décrite par les points O et A . En réalité, la véritable position de P est le point B .

Sous ces hypothèses, l'erreur commise est la distance du point B à la droite $\mathcal{D}$.

Chapitre 3. Calibrage intrinsèque des caméras

Dans le repère $(O, \vec{x}, \vec{y}, \vec{z})$, on a :

$$O = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad A = \begin{pmatrix} x_A \\ l \\ d \end{pmatrix} \quad B = \begin{pmatrix} x_A \\ 2l \cos \alpha - l \\ d + 2l \sin \alpha \end{pmatrix}$$

$$\text{et} \quad \forall P \in \mathcal{D}, P = \lambda \begin{pmatrix} x_A \\ l \\ d \end{pmatrix}$$

Alors, le point de $\mathcal{D}$ le plus proche de B est défini par :

$$\lambda = 1 + \frac{2l(l(\cos \alpha - 1) + d \sin \alpha)}{x_A^2 + l^2 + d^2} = 1 + \frac{N}{D}$$

et l'erreur e par :

$$e = \sqrt{\frac{-N^2 + 8Dl^2(1 - \cos \alpha)}{D}}$$

Pour $x_A = l = 765/2 \text{ mm} \approx 380 \text{ mm}$ (c'est-à-dire pour un point situé dans un coin de la mire) et une distance $d = 2.5 \text{ m}$, les erreurs engendrées par différents angles, sont les suivantes :

angle	erreur
5.0°	16.05 mm
3.0°	9.11 mm
1.0°	2.87 mm
0.5°	1.41 mm

Ainsi, même pour des angles relativement faibles, l'erreur engendrée n'est pas négligeable. Nous sommes en train d'essayer d'estimer la valeur de l'angle α de notre système, afin de nous faire une idée de l'erreur ainsi engendrée.

D'autre part, cette erreur n'est pas intégrée par la méthode d'évaluation d'erreur décrite précédemment. En effet, le fait que les plans de test soient parallèles aux plans de calibrage "absorbe" les erreurs engendrées par l'hypothèse abusive de perpendicularité (voir figure 3.10).

On décide d'utiliser les mesures de l'Optotrak, même si elles n'ont qu'une piètre précision, car les erreurs ainsi engendrées sont mesurables. En effet, elles sont intégrées par la méthode d'estimation d'erreur décrite précédemment, contrairement aux erreurs engendrées par l'hypothèse abusive de perpendicularité.

3.2.3.3 Expérimentations

Première expérience : calibrage à focale fixe par la méthode des deux plans avec une modélisation par splines bicubiques

Pour ce premier test, on fixe la focale à une certaine valeur et on la laisse fixe tout le long de l'acquisition des images de calibrage et de test.

On mesure l'erreur dans l'espace en millimètres (distance entre le point théorique et la droite de rétroprojection calculée).

Les résultats sont décrits dans le tableau suivant. Les premiers résultats sont ceux obtenus en utilisant les positions du plan de la mire fournies par l'Optotrak. Les résultats exprimés entre parenthèses ont été obtenus en supposant un déplacement exactement perpendiculaire au plan de la mire.

	erreur moyenne	erreur maximale
Plan test $\mathcal{P}_1$	0.27 (0.17)	0.59 (0.40)
Plan test $\mathcal{P}_3$	0.13 (0.17)	0.32 (0.34)
Plan test $\mathcal{P}_5$	0.66 (0.23)	1.41 (0.56)

L'expérience utilisant l'Optotrak ne fournit pas de très bons résultats en particulier pour le plan $\mathcal{P}_5$ (jusqu'à 1.5 mm d'erreur). Cependant ces résultats sont tout-à-fait honorables compte tenu de l'imprécision de l'Optotrak, d'autant plus que le plan défaillant se trouvait à la limite du champ de l'Optotrak.

L'expérience n'utilisant pas l'Optotrak semble fournir des résultats meilleurs. Mais il ne faut pas négliger les erreurs dues à l'hypothèse abusive de perpendicularité, dont l'amplitude dépend de l'angle entre la normale au plan de la mire et l'axe de la translation.

Notons que nous avons utilisé les mêmes données pour tester la méthode des deux plans avec une modélisation polynômiale cubique et que les résultats obtenus étaient tout-à-fait similaires. Ceci confirme qu'une modélisation avec des polynômes de degré 3 est suffisante.

Deuxième expérience : calibrage à focale identique par la méthode des deux plans avec une modélisation par splines bicubiques.

Cette fois, la focale est fixée lors de l'acquisition des 2 plans de calibrage, mais elle est modifiée avant d'être ramenée à la même valeur pour l'acquisition des images de test. Notons que nous avons fait des tests pour une focale de $f_0 = 504$ pas codeurs, en amenant le zoom à cette position toujours à partir de la même position $f_1 = 700$ pas codeurs. La mire, le module de camescope et l'Optotrak sont placés selon la même configuration que lors de l'expérience précédente. Les résultats obtenus sont les suivants :

	erreur moyenne	erreur maximale
Plan test $\mathcal{P}_1$	0.30 (0.23)	0.75 (0.57)
Plan test $\mathcal{P}_3$	0.19 (0.28)	0.63 (0.59)
Plan test $\mathcal{P}_5$	0.62 (0.32)	1.46 (0.79)

On constate que ces résultats sont légèrement moins bons que ceux obtenus en maintenant le zoom à une position fixe. Néanmoins, la faible différence entre ces

Chapitre 3. Calibrage intrinsèque des caméras

deux expériences justifie l'utilisation des modules de comescope même en modifiant la focale. Cependant, celle-ci doit être commandée de telle sorte que la position de départ soit à peu près identique à celle utilisée lors du calibrage.

Troisième expérience : calibrage à focale variable par la méthode des deux plans avec une modélisation par splines tricubiques.

La focale du module de comescope est commandée à partir d'une valeur supérieure de 30 pas codeur, c'est-à-dire que pour l'envoyer à une valeur f , on l'envoie d'abord à la valeur $f + 30$, puis à la valeur f , afin d'obtenir une meilleure répétabilité. Tout au long des tests (pour l'acquisition des images dans les plans de calibrage et dans les plans de tests), nous avons bloqué la mise au point à une valeur choisie par l'autofocus pour un plan intermédiaire des deux plans de calibrage (en l'occurrence le plan $\mathcal{P}_3$ de la figure 3.4).

Nous avons utilisé 12 focales différentes pour effectuer le calibrage. Les valeurs sont les mêmes dans les deux plans de calibrage. Elles sont réparties régulièrement entre 504 et 838 pas codeur ($f = 504, 537, 567, 595, 624, 656, 687, 715, 747, 776, 804$ et 838), ce qui, selon une estimation par un modèle à sténopé, correspond à peu près à des valeurs $\alpha_u \approx \alpha_v^5$ comprises entre 5000 et 2400.

Pour chaque plan de test, nous avons acquis 11 images pour des valeurs de focales intermédiaires de celles utilisées lors du calibrage ($f = 521, 552, 580, 613, 641, 669, 704, 732, 761, 789$ et 823). Pour chacune d'elles, nous avons évalué la distance maximale et la distance moyenne entre la position 3D théoriques des points et la droite de rétroprojection associée à leur image. On peut voir les résultats dans le tableau 3.1.

Pour ne pas biaiser les valeurs moyennes de l'erreur moyenne et de l'erreur maximale en donnant un poids plus fort aux faibles focales pour lesquelles le nombre de points présents dans l'image est plus important, nous n'avons pas calculé ces valeurs sur l'ensemble des points dont nous disposons dans l'ensemble des images, mais nous avons d'abord calculé l'erreur moyenne pour chaque valeur de focale sur l'ensemble des points présents dans l'image correspondante, puis nous avons calculé la moyenne de ces erreurs moyennes.

On peut constater que les résultats obtenus avec le système de repérage mécanique sont en moyenne comparables à ceux obtenus pour une focale fixe, ce qui montre que les splines s'adaptent bien aux données (la focale $f = 613$ constituant une exception que nous ne sommes pas parvenus à expliquer). En revanche, les résultats obtenus avec le localisateur optotrak sont moins bons. Cependant ces erreurs sont tout-à-fait cohérentes avec l'évaluation que nous avons faite des erreurs engendrées par l'utilisation de l'optotrak.

Pour un champ couvert d'environ $800 \times 700 \times 700 \text{ mm}^3$, nous obtenons une erreur moyenne de 1.2 mm , ce qui est inférieur à 0.2% , et une erreur maximale de 2.8 mm , ce qui représente 0.4% .

5. Rappelons que $\alpha_u = \frac{F}{k_u}$ et $\alpha_v = \frac{F}{k_v}$ où F est la longueur focale et $k_u \times k_v$ est la taille d'un pixel. L'ordre de grandeur de k_u et de k_v est de 0.01 mm/pixel .

3.3 Conclusion

Dans le cadre d'un calibrage à focale fixe, les méthodes multi-plans sont plus souples que les méthodes dérivées d'un modèle à sténopé dans la mesure où elles n'imposent pas la concurrence des droites de rétroprojection.

L'interpolation des données peut se faire de manière plus ou moins complexe, le recours à des modèles locaux (splines) n'étant pas forcément indispensable.

Ainsi nous avons retenu la méthode des deux plans avec une interpolation polynomiale cubique qui offre un bon compromis de précision et de vitesse d'exécution. Nos expérimentations nous permettent de dire que la précision obtenue est en moyenne de 0.35 mm pour un champ observé d'environ $800 \times 700 \times 700 \text{ mm}^3$, ce qui représente 0.05% . Ces résultats nous satisfont pleinement.

Dans le cadre d'un calibrage à focale variable, nous avons proposé une nouvelle méthode de calibrage : la méthode des deux plans avec modélisation par splines tricubiques. Cette approche est une extension de la méthode multi-plans avec une modélisation par splines bicubiques jusqu'à présent développées pour une focale fixe. La précision évaluée lors de nos expérimentation est en moyenne de 1.2 mm pour un champ observé d'environ $800 \times 700 \times 700 \text{ mm}^3$, ce qui représente 0.2% . On a donc perdu en précision par rapport à un calibrage à focale fixe. Néanmoins nos expérimentations prouvent qu'une caméra à focale variable est exploitable pour faire de la mesure.

Les méthodes que nous avons retenues utilisent deux plans de calibrage. On a vu que la rétroprojection est aisée à calculer. En revanche l'obtention d'une projection précise n'est pas directe. Dans la suite, nous raisonnerons donc souvent dans l'espace (mesure de distance point 3D / droite de rétroprojection) plutôt que dans l'image (mesure de distance point 2D / point 2D) contrairement à ce qui est habituellement fait par les gens qui utilisent des modèles dérivés du modèle à sténopé.

Chapitre 3. Calibrage intrinsèque des caméras

TAB. 3.1 - Erreurs de calibrage d'une caméra à focale variable : repérage par l'Op-
totrak, et entre parenthèses, repérage par le système mécanique.

		erreur moyenne	erreur maximale
Plan test $\mathcal{P}_1$	$f = 521$	0.60 (0.16)	0.76 (0.39)
	$f = 552$	0.63 (0.14)	0.76 (0.29)
	$f = 580$	0.68 (0.27)	0.98 (0.50)
	$f = 613$	0.64 (0.33)	1.00 (0.61)
	$f = 641$	0.64 (0.11)	0.75 (0.18)
	$f = 669$	0.63 (0.19)	0.87 (0.41)
	$f = 704$	0.61 (0.17)	0.78 (0.31)
	$f = 732$	0.65 (0.18)	0.85 (0.35)
	$f = 761$	0.62 (0.20)	0.81 (0.37)
	$f = 789$	0.66 (0.15)	0.76 (0.23)
	$f = 823$	0.68 (0.13)	0.76 (0.21)
	valeur moyenne	0.64 (0.19)	0.83 (0.35)
	valeur maximale	0.68 (0.33)	1.00 (0.61)
Plan test $\mathcal{P}_3$	$f = 521$	1.10 (0.18)	1.76 (0.45)
	$f = 552$	1.08 (0.15)	1.66 (0.38)
	$f = 580$	1.06 (0.25)	1.80 (0.54)
	$f = 613$	1.16 (0.41)	1.74 (0.75)
	$f = 641$	1.10 (0.17)	1.57 (0.40)
	$f = 669$	1.05 (0.17)	1.46 (0.33)
	$f = 704$	1.11 (0.22)	1.54 (0.48)
	$f = 732$	1.13 (0.15)	1.59 (0.37)
	$f = 761$	1.18 (0.27)	1.62 (0.49)
	$f = 789$	1.14 (0.16)	1.43 (0.29)
	$f = 823$	1.11 (0.16)	1.42 (0.25)
	valeur moyenne	1.10 (0.21)	1.60 (0.43)
	valeur maximale	1.18 (0.41)	1.80 (0.75)
Plan test $\mathcal{P}_5$	$f = 521$	1.79 (0.37)	2.54 (0.70)
	$f = 552$	1.79 (0.30)	2.41 (0.50)
	$f = 580$	1.78 (0.36)	2.35 (0.78)
	$f = 613$	1.86 (0.54)	2.81 (1.17)
	$f = 641$	1.81 (0.29)	2.43 (0.50)
	$f = 669$	1.80 (0.28)	2.34 (0.60)
	$f = 704$	1.89 (0.30)	2.51 (0.56)
	$f = 732$	1.78 (0.29)	2.13 (0.53)
	$f = 761$	1.87 (0.30)	2.37 (0.48)
	$f = 789$	1.93 (0.24)	2.25 (0.36)
	$f = 823$	1.93 (0.27)	2.29 (0.65)
	valeur moyenne	1.82 (0.34)	2.40 (0.62)
	valeur maximale	1.93 (0.54)	2.81 (1.17)

Chapitre 4

Recalage des caméras par rapport à l'environnement

On a vu au chapitre précédent que le calibrage intrinsèque des caméras s'effectue généralement grâce à l'intermédiaire d'une mire. Ainsi les résultats sont obtenus par rapport à un référentiel 3D associé à cette mire. Il faut ensuite recaler ces résultats par rapport à un repère extérieur intéressant, associé à l'environnement de travail. Dans le cas de l'application nucléaire, il s'agit d'un repère associé au bol du générateur de vapeur dans lequel on va observer les mouvements du robot. Dans le cas de l'application médicale, on va se rattacher au repère associé à la base du robot qui opère.

On peut se reporter au paragraphe 8.1.1 pour une justification du choix d'un repère absolu spécifique à chaque application. Dans la suite, nous allons présenter l'approche retenue pour chacune de nos deux applications.

4.1 Méthode proposée pour l'application nucléaire

4.1.1 Rappel des besoins et conditions de travail

L'environnement de travail est un bol de générateur de vapeur. Le plafond de ce bol est constitué d'une plaque tubulaire.

L'objectif du projet consiste à vérifier que le robot est bien positionné par rapport au tube qu'il doit inspecter. On oriente donc les caméras en direction du plafond et on choisit comme référentiel absolu, un repère associé à la plaque tubulaire, d'autant plus que celle-ci constitue une mire naturelle qui va faciliter le recalage des caméras (figure 4.1).

Il s'agit de calculer la matrice de passage T entre R_{cam} et le repère absolu, en l'occurrence R_{bol} .

Soit $(P)_{R_{bol}}$, un point 3D dont les coordonnées sont exprimées dans R_{bol} . Soit $(P)_{R_{cam}}$, le même point dont les coordonnées sont exprimées dans R_{cam} . On a alors : $(P)_{R_{bol}} = T.(P)_{R_{cam}}$

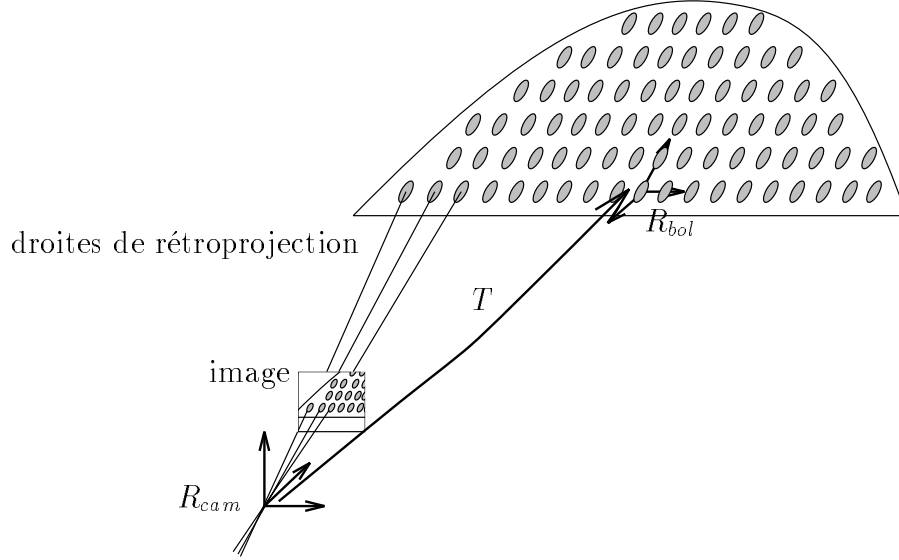


FIG. 4.1 - Recalage par rapport à la plaque tubulaire du bol de générateur de vapeur

Considérons maintenant les centres des trous de la plaque tubulaire. On connaît leurs coordonnées dans R_{bol} par construction du bol : soit $(P_i)_{R_{bol}}$, le i^{me} point visible par la caméra. On peut traiter l'image renvoyée par la caméra pour détecter les coordonnées (u_i, v_i) de ce point dans l'image. Le calibrage de caméra étant effectué, on sait associer à tout pixel (u_i, v_i) , une droite de rétroprojection $\mathcal{D}_i$ exprimée dans le repère R_{cam} . En théorie, le point (P_i) appartient à $\mathcal{D}_i$. On doit donc trouver T telle que pour tout i , $T^{-1} \cdot (P_i)_{R_{bol}}$ appartient à $(\mathcal{D}_i)_{R_{cam}}$, ou du moins telle qu'elle minimise D :

$$D = \sum_i d^2 \left(T^{-1} \cdot (P_i)_{R_{bol}}, (\mathcal{D}_i)_{R_{cam}} \right) \quad (4.1)$$

4.1.2 Etude bibliographique

Dans la littérature, le problème décrit dans le paragraphe 4.1.1, n'est généralement pas abordé de cette manière. Il a cependant déjà été traité sous des formes équivalentes. Des conditions d'existence et d'unicité de la solution ont été mises en exergue et divers types de méthodes de résolution ont été élaborées.

A notre connaissance, les études menées à ce sujet ont toujours été faites avec un modèle de caméra à sténopé qui suppose la concourance des droites de rétroprojection.

4.1.2.1 Problèmes équivalents

Notre objectif consiste à recalibrer par rapport à l'environnement, la caméra déjà calibrée par rapport à une mire, ce qui revient à déterminer la matrice de passage

T entre les repères R_{abs} et R_{cam} (cf figure 4.1, où le repère absolu est le repère R_{bol}). Dans le cas de l'utilisation d'un modèle à sténopé, il s'agit de déterminer les paramètres extrinsèques de la caméra lorsque les paramètres intrinsèques sont connus.

Un problème très similaire consiste à retrouver la position et l'orientation d'un objet (auquel est associé un repère R_{obj}) dans le repère R_{cam} de la caméra qui l'observe, grâce à des points ou segments caractéristiques de cet objet. Dans ce cas, il s'agit de déterminer la matrice de passage entre les repères R_{cam} et R_{obj} , c'est-à-dire l'inverse de la matrice précédente.

Les indices utilisés pour la résolution de ce type de problème peuvent être des points, comme c'est décrit au paragraphe précédent. Cependant, les solutions généralement proposées ont recours à des droites, ce qui n'est pas une contrainte, puisqu'il suffit d'associer deux points pour disposer d'une droite. L'utilisation de droites plutôt que de points permet de découpler l'estimation de la rotation et celle de la translation [LHF90]. On commence par estimer la meilleure rotation. Une fois la rotation optimale déterminée, trouver la translation est un problème linéaire.

4.1.2.2 Méthodes itératives

Ces méthodes sont générales et ne dépendent pas de la configuration des données (les données sont en l'occurrence les coordonnées des points de l'espace et de leurs images). Il s'agit de résoudre un système d'équations non linéaires. Le nombre d'équations étant généralement grand par rapport au nombre d'inconnues, on cherche à minimiser une fonction d'erreur.

La formulation de la fonction d'erreur et le choix de la procédure de minimisation impliquent différentes méthodes de résolution.

La fonction d'erreur vise à traduire le plus ou moins bon respect de certains critères sur l'ensemble des données : elle est donc la somme, éventuellement pondérée, de la quantification de ces critères en chacun des couples de données. Ces critères peuvent être la distance au carré (idéalement nulle) entre un point 3D et la droite de rétroprojection associée à son correspondant dans l'image (cf paragraphe 4.1.1) ou bien la distance au carré (idéalement nulle) entre la projection d'un point et son correspondant dans l'image [Low91] ou bien le produit scalaire (idéalement nul) de 2 vecteurs qui traduit leur perpendicularité [HM93].

La représentation de la rotation peut également avoir une incidence sur la fonction d'erreur. Si cette représentation est minimale (3 paramètres), telle qu'avec les angles d'Euler [LHF90], la fonction d'erreur n'est pas modifiée. En revanche, si la représentation de la rotation compte plus de 3 paramètres, la fonction d'erreur doit comporter un terme supplémentaire qui traduise les contraintes entre ces paramètres (norme unitaire pour une représentation par un quaternion [HM93], contraintes d'orthonormalité pour une représentation par une matrice 3x3 [Yua89]).

La minimisation s'effectue souvent selon une méthode de Newton [Low91, Yua89].

Chapitre 4. Recalage des caméras par rapport à l'environnement

[HM93] utilise la méthode de région de confiance qui consiste à approximer la fonction d'erreur par une forme quadratique locale, dans un voisinage de la solution courante.

[FB81] adopte une stratégie complètement différente. Une première estimation des paramètres est effectuée à partir d'un ensemble minimal de données. Seules les données cohérentes avec cette estimation sont conservées. Si ces données sont en nombre suffisant, elles servent à effectuer une nouvelle estimation plus fine, selon un procédé classique (des moindres carrés par exemple). Cette méthodologie permet de s'affranchir des données aberrantes.

L'inconvénient majeur des méthodes itératives est bien sûr la présence de minima locaux, inhérents aux problèmes d'optimisation non linéaires. Une bonne initialisation des paramètres s'avère donc indispensable. Les méthodes de détermination directe des paramètres à partir d'un nombre restreint de données répondent bien à cet impératif.

4.1.2.3 Méthodes directes

Pour un nombre de correspondances image/espace restreint, on peut trouver des solutions exactes en résolvant un système d'équations non linéaires.

L'équipe de M. Dhome [DRLR89] cherche la transformation qui permet de considérer un triplet de droites de l'image comme la projection d'un triplet de droites de l'espace.

Chen [Che91a] cherche à faire correspondre des droites et des plans exprimés dans des repères différents dans l'espace. Notre problème entre dans le cadre de son étude si l'on prend comme droites, les droites constituées à partir de points caractéristiques de l'objet observé, et comme plans, les plans d'interprétation décrits figure 4.2. Les droites sont connues dans le repère R_{abs} alors que les plans sont calculés dans le repère R_{cam} à partir de l'image des points caractéristiques et des paramètres intrinsèques de la caméra.

[DRLR89] et [Che91a] aboutissent à une équation de degré 8 ne comportant qu'une seule inconnue. Cette équation est résolue par des méthodes numériques itératives. Des configurations particulières permettent de diminuer le degré de l'équation (les 3 droites-espace sont orthogonales, les 3 droites-espace sont coplanaires, au moins 2 droites-espace sont parallèles, intersection des 3 droites-espaces en un même point). Notons que dans le cas particulier qui nous intéresse (où les plans considérés sont des plans d'interprétation), l'équation de Chen est déjà simplifiée.

[FB81] cherche la transformation qui permet de considérer n points de l'image comme étant la projection de leurs homologues dans l'espace. n peut varier de 3 à 6. Les contraintes de distance entre les points sont à la base du raisonnement et impliquent 2 équations, l'une de degré 4 (qui peut être résolue de manière explicite ou itérative), l'autre de degré 2.

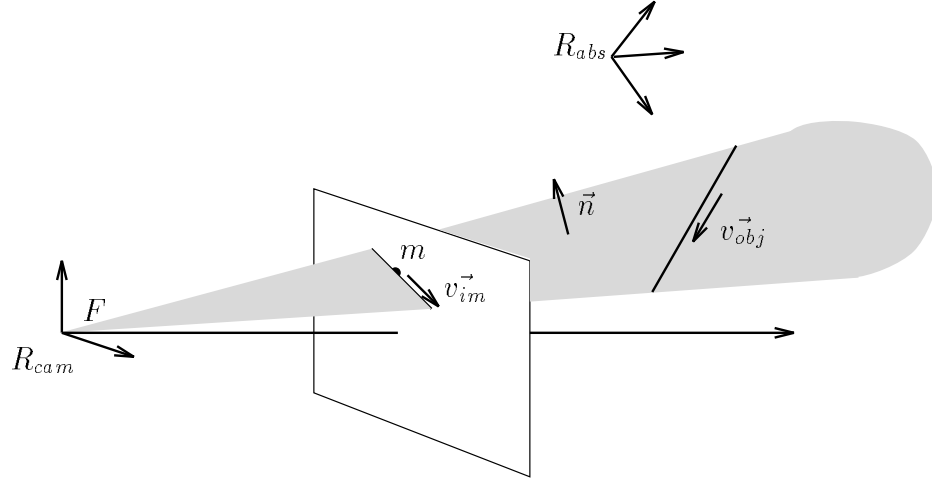


FIG. 4.2 - Plan d'interprétation défini par le centre de projection F et la droite image de vecteur directeur $\vec{v}_{im}$

4.1.2.4 Conditions d'existence d'une solution

La résolution des équations évoquées précédemment a conduit Chen [Che91a] à étudier les conditions d'existence de ces solutions. Il en arrive aux conclusions suivantes :

- La rotation ne peut pas être déterminée si et seulement si l'une des trois conditions suivantes est vérifiée (le rôle des droites et des plans étant interchangeable) :
 - 2 des droites sont parallèles ainsi que les 2 plans correspondants
 - 2 des droites sont parallèles et sont perpendiculaires à la troisième, et les 2 plans correspondant aux droites parallèles sont perpendiculaires au troisième plan
 - les trois droites ou les trois plans sont parallèles
- La translation ne peut pas être déterminée si et seulement si les normales aux trois plans sont linéairement dépendantes, ce qui signifie que les trois plans ne doivent pas être parallèles à une même droite de l'espace.

En étudiant le problème de manière algébrique, Yuan [Yua89] en déduit une condition suffisante d'existence d'une solution pour la rotation : une solution générique existe si $3(r + 1) - 2n > 0$ où n est le nombre de correspondances points-espace/points-image et r est la dimension du sous-espace linéaire engendré par les n points-espace. On en déduit qu'au moins une solution exacte existe lorsqu'on dispose de :

- 1 seul point ($\{n, r\} = \{1, 0\}$)

- 2 points ($\{n, r\} = \{2, 1\}$)
- 3 ou 4 points non colinéaires ($\{n, r\} = \{3, 2\}, \{4, 2\}$ ou $\{4, 3\}$)
- 5 points non coplanaires ($\{n, r\} = \{5, 3\}$)

Dans les autres cas (par exemple si l'on dispose de 6 points ou plus), le système peut être surcontraint si les données ne sont pas cohérentes, auquel cas il faut utiliser les méthodes itératives de minimisation décrites précédemment.

4.1.2.5 Unicité de la solution

Lorsque le nombre de points est inférieur à trois, il est clair qu'on dispose d'une infinité de solutions.

Dans le cas de 3 points, [FB81] a démontré qu'il existe au plus 4 solutions et fournit des exemples numériques et graphiques de solutions multiples dans le cas de 3 points et de 4 ou 5 points non coplanaires.

[FB81] démontre également de manière analytique que la configuration constituée de 4 points coplanaires admet une solution unique, ainsi que les configuration comportant au moins 6 points. Yuan [Yua89] démontre de manière algébrique que la configuration des 4 points coplanaires est la seule pour laquelle il existe une et une seule solution.

Dans les autres cas, il existe plusieurs solutions. Cependant certaines d'entre elles peuvent être complexes. D'autre part, on peut avoir recours à des règles simples pour discriminer les solutions réalistes des autres [DRLR89]. Le fait que l'objet doit se trouver devant la caméra est un exemple de telles règles.

4.1.2.6 Précision

Yuan a étudié la précision de sa méthode [Yua89] et a constaté qu'elle était meilleure dans le cas de points non coplanaires. Les effets sont d'autant plus importants que le plan est parallèle au plan image. Cependant, ils peuvent être atténués par l'influence de la longueur focale : la précision est d'autant meilleure que la longueur focale est importante.

4.1.3 Méthode proposée

Notons que toutes les méthodes mentionnées précédemment font l'hypothèse d'un modèle à sténopé, ce qui n'est pas notre cas (modèle bi-plan). Elles ne sont donc pas applicables directement.

La méthode proposée comporte deux phases :

- Une phase d'initialisation : On rappelle qu'on a proposé d'utiliser un modèle à sténopé en parallèle du modèle des 2 plans, pour la procédure de projection qui ne requiert pas une grande précision. L'utilisation de ce modèle peut nous permettre ici d'estimer grossièrement les paramètres de la matrice de passage, en

appliquant l'une des méthodes décrites précédemment. On retient la méthode de [FB81] qui fournit une solution directe pour 4 points coplanaires (nous ne disposons en effet que de données coplanaires). De plus, la solution est unique dans ce cas. On choisit donc 4 couples point-espace/point-image pour obtenir une estimation grossière des paramètres cherchés, ce qui permet d'initialiser la seconde phase.

- Une phase d'optimisation des paramètres : Cette phase consiste à minimiser D , la somme des distances au carré entre une droite de rétroprojection et le point 3D correspondant (voir formule 4.1). Cette fois-ci, les droites de rétroprojection sont calculées selon le modèle choisi, en l'occurrence un modèle bi-plan, et l'on considère l'ensemble des données dont on dispose. Cette minimisation peut se faire selon une méthode itérative classique ([PFTV92] chapitres 10 et 14) de type descente de gradient, Newton, Levenberg-Marquardt

En pratique, pour la phase d'initialisation, on choisit les 4 correspondances telles que les points images soient répartis dans les 4 coins de l'image, à une distance au bord comprise entre 20 et 100 pixels ; pour la phase d'optimisation, on utilise la méthode des gradients conjugués [PFTV92].

4.1.4 Résultats

Nous avons expérimenté la méthode de [FB81] sur des données simulées. Elle fournit d'excellents résultats. En revanche, sur des données réelles, elle fournit une matrice de transformation aberrante. Nous n'avons pas encore testé la méthode sur des données simulées bruitées pour étudier son comportement en présence de bruit.

Cette première phase d'initialisation ne fonctionnant pas, nous avons utilisé la phase d'optimisation seule sur des données initiales arbitraires. En raison des nombreux minima locaux, cette manière de procéder aboutit rarement aux paramètres recherchés. Nous discrétisons donc l'ensemble des paramètres possibles et nous lançons l'algorithme d'optimisation sur ces différentes données initiales. Nous retenons la solution qui fournit les résidus les plus faibles et nous réitérons l'opération sur un nouvel ensemble de paramètres possibles répartis autour de la solution retenue à l'étape précédente. Cette manoeuvre est répétée quatre fois et nous fournit finalement une erreur moyenne entre les points 3D et leur droite associée inférieure à 1.0 *mm* et une erreur maximale inférieure à 1.5 *mm* dans la majorité des cas.

Nous avons également mis en place une expérimentation pour comparer la matrice obtenue par notre algorithme de recalage avec une matrice fournie par un autre système de mesure, en l'occurrence l'Optotrak. La configuration du matériel est décrite sur le schéma 4.3.

On calibre la caméra en repérant les plans de calibrage dans le repère associé à l'optotrak (voir la partie concernant le repérage au moyen de l'Optotrak au paragraphe 3.2.3.2). Le repère R_{mire} est donc identifié au repère $R_{optotrak}$.

On déplace ensuite la mire de calibrage. Sa nouvelle position est repérée par l'Optotrak grâce au "rigid body" fixé sur la mire. La matrice de passage ainsi obte-

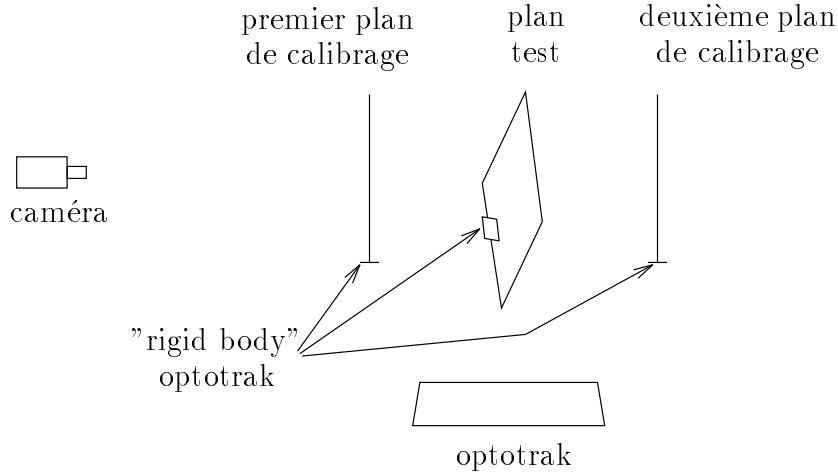


FIG. 4.3 - *Principe de la vérification du recalage de caméra*

nue est théoriquement l'inverse de la matrice que nous fournit notre algorithme de recalage de caméra.

Nous multiplions ces deux matrices pour obtenir une matrice P . P devrait être la matrice identité, mais est bien sûr légèrement différente. Pour évaluer cette différence, nous analysons la matrice P selon ses composantes en translation et nous extrayons l'axe et l'angle de la rotation selon la méthode exposée dans [Aya89] p. 222.

Nous avons effectué cette vérification sur 5 positions différentes de la mire. Pour des raisons pratiques, la mire est toujours restée dans un plan presque vertical et n'a été déplacée que d'un angle compris entre -20° et $+20^\circ$. Les résultats obtenus sont les suivants :

- La matrice P a un angle de rotation moyen de 0.8° et toujours inférieur à 1.4° .
- Ses translations selon les axes x et y sont du même ordre de grandeur, à savoir environ 1.5 mm en moyenne et 3 mm au maximum.
- Sa translation selon l'axe optique de la caméra est bien moins bonne puisqu'elle est de 17 mm en moyenne et de 31 mm au maximum.

4.2 Méthode proposée pour l'application médicale

4.2.1 Rappel des besoins et conditions de travail

L'espace de travail à surveiller n'est pas très grand ($30 \times 30 \times 30\text{ cm}^3$). On ne dispose pas, comme dans l'application nucléaire, de repères naturels auxquels

se référer. Dans le but d'être indépendant des divers systèmes d'imagerie qu'il est possible d'utiliser en tant que localisateur pour les applications médicales, on a choisi de se recalculer par rapport au robot lui-même plutôt que par rapport au système d'imagerie.

4.2.2 Méthode proposée

L'idée consiste à calibrer directement les caméras dans le repère qui nous intéresse, à savoir le repère du robot. Pour cela, on va lier matériellement la mire de calibrage au robot. Le repère absolu de calibrage que l'on a nommé jusqu'à présent R_{mire} est donc confondu avec R_{robot} .

Le seul pré-requis nécessaire est que le robot soit calibré.

D'autre part, on doit disposer d'un localisateur 3D. Les expérimentations qui suivent ont été effectuées avec le système Optotrak.

Enfin, la mire utilisée présente les caractéristiques suivantes :

- Elle peut être fixée sur le poignet du robot.
- On peut fixer sur elle un "rigid body" optotrak (figure 4.4).

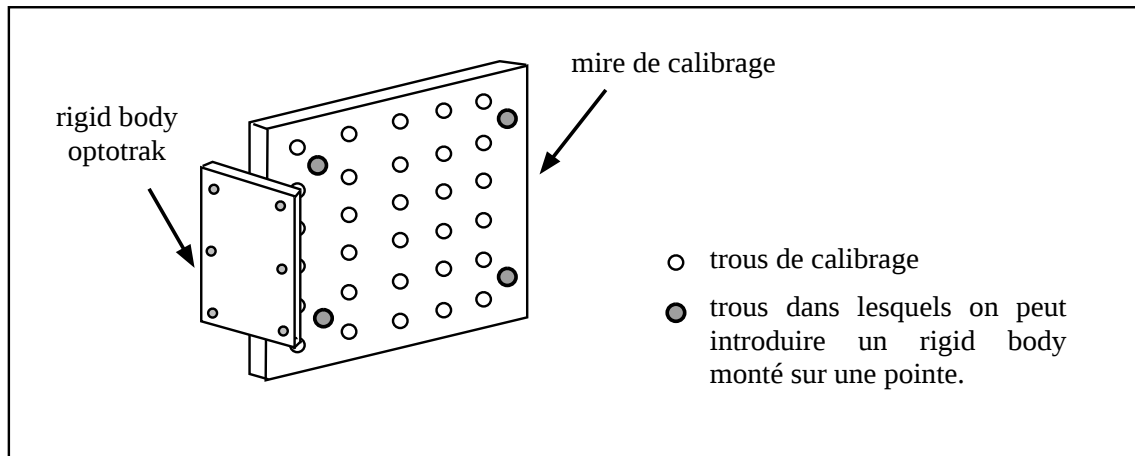


FIG. 4.4 - Mire de calibrage utilisée pour l'application médicale

- Elle mesure $226 \times 212 \text{ mm}^2$. Elle est en plexiglas opaque, pour assurer une meilleure diffusion de la lumière, et est recouverte d'une pellicule noire elle-même percée de 144 trous de 3 mm de diamètre, dont l'emplacement est connu avec une précision de 0.01 mm dans un repère que l'on nommera R_{plaque} .
- Elle comporte 4 trous dans lesquels on peut introduire une pointe montée sur un rigid body optotrak. Cette pointe est équipée d'un anneau qui permet de bloquer le rigid body contre la mire (figure 3.8). La position de ces 4 trous est connue avec une précision de 0.01 mm dans le repère R_{plaque} .

Méthodologie

On fixe la mire sur le poignet du robot (figure 4.5).

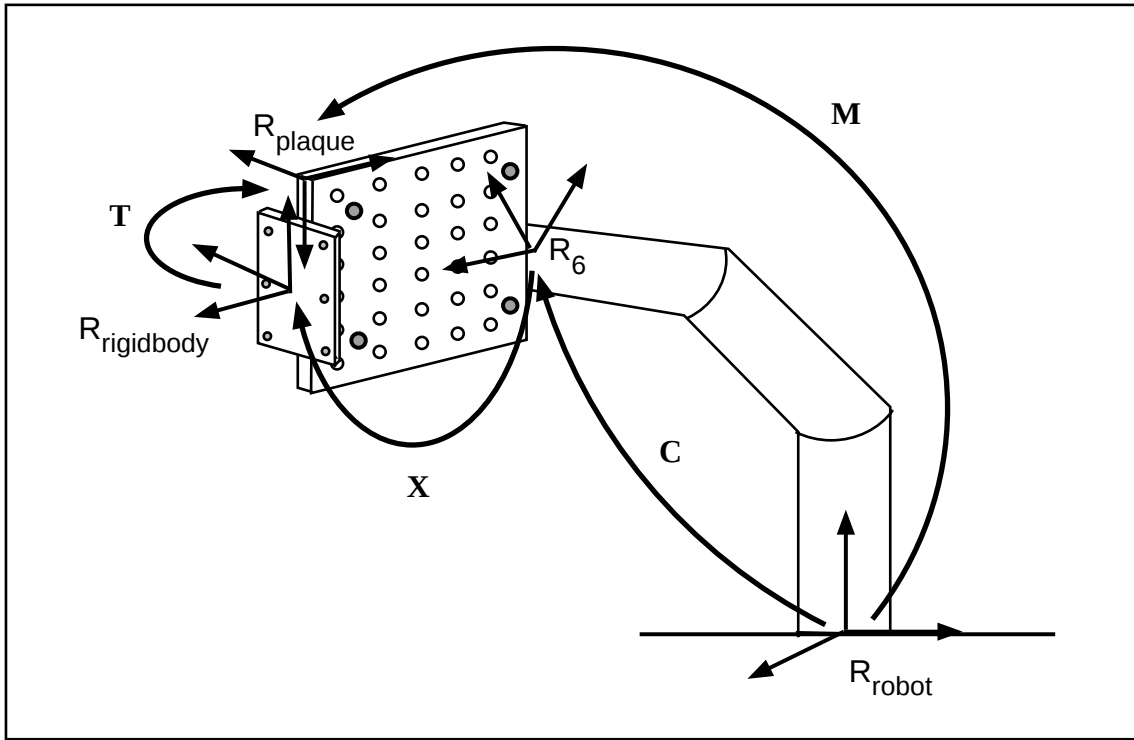


FIG. 4.5 - Recalage par rapport au robot

On cherche à déterminer M , la matrice de passage entre R_{plaque} et R_{robot} . M est la combinaison des matrices C , X et T .

- C , qui dépend de la configuration du robot, est obtenue par le changeur de coordonnées.
- On détermine X , qui est indépendante de la configuration du robot, par la résolution d'une série d'équations $A_i \cdot X = X \cdot B_i$ (méthodologie identique à celle du calibrage des marqueurs, voir chapitre 5)
- On détermine T , qui est également indépendante de la configuration du robot, par l'intermédiaire du rigid body supplémentaire muni d'une pointe annelée (voir photo 4.6). On procède de la même manière qu'au paragraphe 3.2.3.2.

Une fois les matrices X et T déterminées, on procède au calibrage de la caméra :

- On amène le robot dans une configuration telle que la mire soit approximativement perpendiculaire à l'axe optique de la caméra

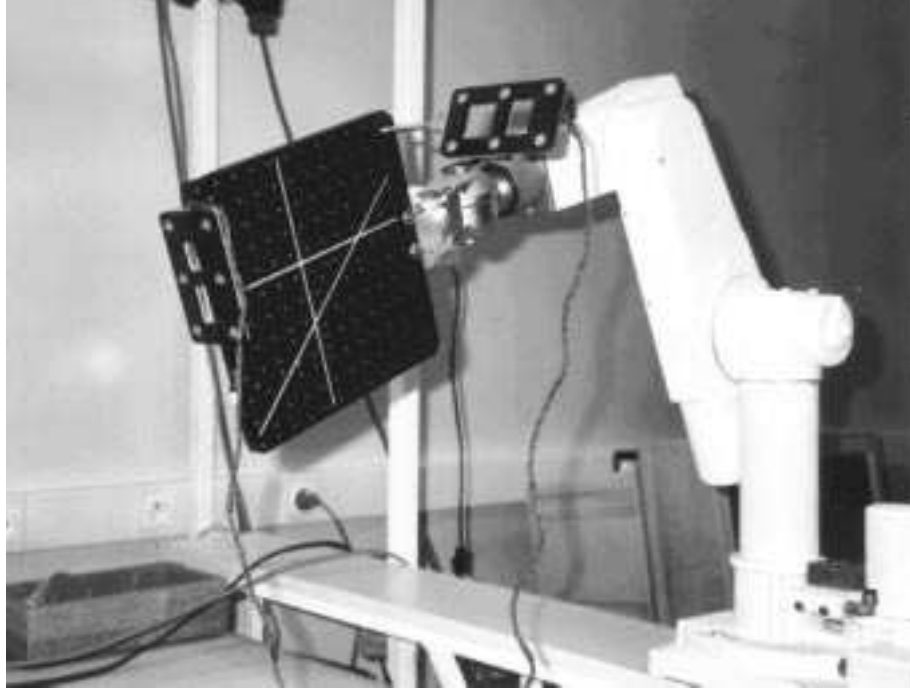


FIG. 4.6 - Détermination de T au moyen d'un "rigid body" supplémentaire, muni d'une pointe annelée

- On récupère les valeurs articulaires du robot $(q_{1i})_{i=1..6}$, ce qui nous permet de connaître la matrice C_1 grâce au changeur.

La combinaison des matrices C_1 , X et T nous fournit une matrice de passage entre R_{plaque} et R_{robot} : $(P)_{R_{robot}} = M_1 \cdot (P)_{R_{plaque}} = C_1 \cdot X \cdot T \cdot (P)_{R_{plaque}}$.

- On déplace le robot tout en maintenant la plaque de la mire parallèle à sa première position et l'on acquiert les valeurs $(q_{2i})_{i=1..6}$ qui nous fournissent une matrice M_2 : $M_2 = C_2 \cdot X \cdot T$.

On est donc dans la même situation que pour la méthode "classique" de calibrage de caméras avec pour repère absolu R_{robot} , au lieu d'un repère quelconque. Ainsi la droite de rétroprojection correspondant à un couple (u, v) donné, est calculée dans R_{robot} .

4.2.3 Résultats

Le matériel utilisé pour ces tests comprend :

- un robot PUMA 260.
- une caméra JVC munie d'un objectif de 25 mm.
- la mire décrite précédemment.

Chapitre 4. Recalage des caméras par rapport à l'environnement

On effectue le même test qu'au paragraphe 3.2.3 : le robot est amené en deux positions telles que la mire soit approximativement perpendiculaire à l'axe optique de la caméra. Ces deux plans sont utilisés pour le calibrage par la méthode des deux plans avec une modélisation par splines bicubiques. Puis on acquiert trois images de test telles que le plan de la mire soit devant, derrière et entre les deux plans de calibrage. La distance séparant deux plans est approximativement de 6 cm.

Les résultats obtenus sont les suivants :

	erreur moyenne	erreur maximale
Plan test $\mathcal{P}_1$	0.61	1.18
Plan test $\mathcal{P}_3$	0.72	1.26
Plan test $\mathcal{P}_5$	1.11	1.95

La “médiocrité” de ces résultats s'explique par les erreurs engendrées par le robot, d'autant plus que la mire, du fait de son poids, crée une flèche qui influence le comportement du robot.

4.3 Conclusion

La méthode de recalage développée pour l'application nucléaire fournit de bons résultats (distance droite-point moyenne inférieure au millimètre), mais n'est pas optimale en temps d'exécution car la phase d'initialisation destinée à faciliter l'optimisation ne fonctionne pas encore. Il serait intéressant d'étudier l'influence du bruit sur l'algorithme que nous avons choisi, pour nous assurer que c'est bien l'origine du problème. Néanmoins, ce problème n'est pas gênant puisqu'il s'agit d'une phase de calibrage qui ne présente pas d'impératif de temps d'exécution.

La méthode développée pour l'application médicale est une méthode pragmatique qui consiste à calibrer directement les caméras dans le repère associé au robot. L'inconvénient majeur de cette approche est que sa précision dépend de la précision du robot.

D'autre part, dans le cadre du contrôle redondant, cette manière de procéder n'assure pas l'indépendance du système à surveiller - le robot - et du système qui surveille - les caméras. De toutes façons, une totale indépendance n'est pas réalisable car il faut forcément relier les deux systèmes pour exprimer leurs mesures dans un repère commun. Cette approche étant intéressante d'un point de vue pratique car elle supprime une étape de calibrage, c'est celle que nous utiliserons pour effectuer une expérimentation complète du système de contrôle redondant décrite au chapitre 7.

On pourrait utiliser un autre repère absolu dans lequel serait connue la position d'un certains nombres d'indices visuels et procéder alors d'une manière similaire à celle décrite pour l'application nucléaire. Cette manière de procéder nécessiterait une étape supplémentaire, à savoir le calibrage extrinsèque du robot.

Chapitre 5

Le calibrage des marqueurs

Rappelons le problème et la méthodologie générale que nous avons adoptée :

On se place dans un repère absolu (noté R_{abs}). L'objectif final du projet est de vérifier que la position d'un robot surveillé par une caméra est bien celle indiquée par les codeurs angulaires des articulations et souhaitée par l'utilisateur.

Pour cela, on se propose de traquer dans l'image renvoyée par la caméra, des marqueurs apposés sur chacun des segments du robot, c'est-à-dire de vérifier sur l'image que les marqueurs sont là où on les attend. Leur position dépend bien évidemment de la position courante du robot. On a choisi pour le robot un modèle de type Denavit-Hartenberg-Hayati [HTR88]. Les valeurs articulaires et le modèle de changeur de coordonnées dont les paramètres ont été calibrés [Hol89, GQP93] nous permettent de connaître la position et l'orientation des repères liés à chaque segment du robot, notés de manière générale R_{seg} , dans le repère associé au robot, noté R_{robot} . Cependant, il nous faut connaître l'emplacement des marqueurs dans le référentiel absolu.

Un marqueur est fixe par rapport à R_{seg} , le repère attaché au segment qui le supporte. Soit X , la matrice qui permet d'effectuer les changements de coordonnées entre le repère attaché au marqueur, que l'on notera R_{marq} , et R_{seg} .

Le calibrage d'un marqueur consiste à déterminer X (voir figure 5.1). Notons qu'on ne peut pas faire une mesure directe de X , car le repère R_{seg} est un repère abstrait, interne au robot, et n'est donc pas "palpable".

Ce problème est tout-à-fait similaire à celui qui consiste à déterminer la position d'un outil terminal ou d'un capteur porté par le robot, par rapport à R_6 , où R_6 est le référentiel attaché au poignet d'un robot à 6 degrés de liberté tel que le PUMA que nous avons utilisé. L'outil terminal peut être un laser, un capteur de distance, un capteur tactile, une caméra ...

Une série de méthodes ont été proposées pour identifier X : elles sont regroupées sous le terme générique de "calibrage caméra/pince" (ou "hand/eye calibration") car il s'agit la plupart du temps de déterminer la matrice X entre le référentiel associé à la pince du robot et le référentiel associé à une caméra tenue par la pince.

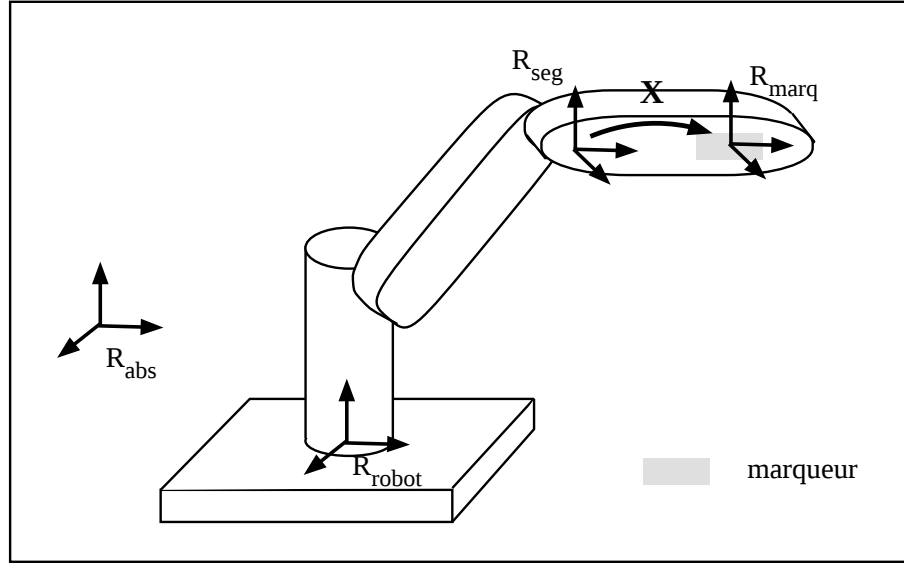


FIG. 5.1 - Problème : Déterminer X

5.1 Objectif : résoudre $A.X = X.B$

5.1.1 Description du problème

L'idée directrice est de considérer l'ensemble R_{seg} / R_{marq} dans diverses positions. Dans le cas de deux positions différentes (voir figure 5.2), le problème peut se poser ainsi :

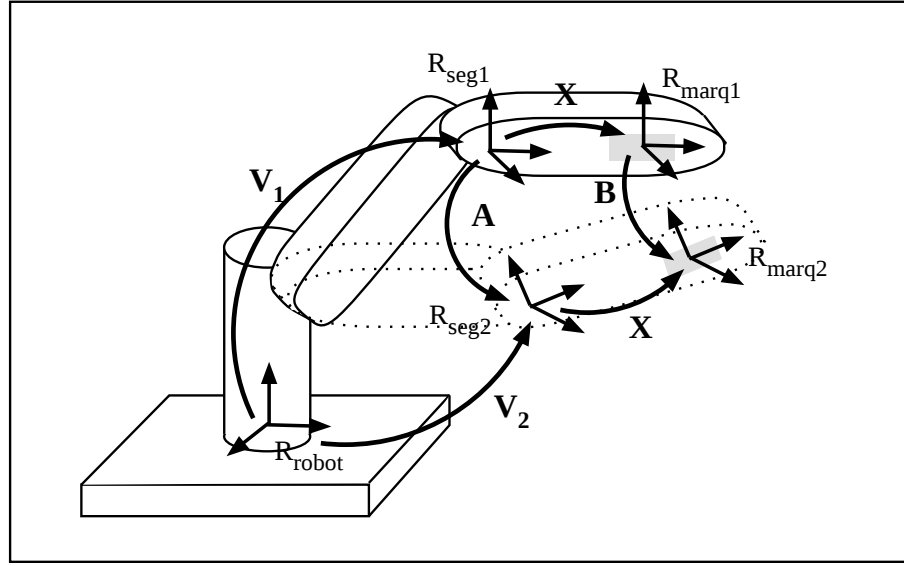
- A représente le changement de coordonnées entre R_{seg} dans la position 2 et R_{seg} dans la position 1 ;
- B représente le changement de coordonnées entre R_{marq} dans la position 2 et R_{marq} dans la position 1.
- D'après la figure 5.2, on voit clairement qu'il s'agit de résoudre l'équation matricielle $A.X = X.B$.

5.1.2 Détermination de A

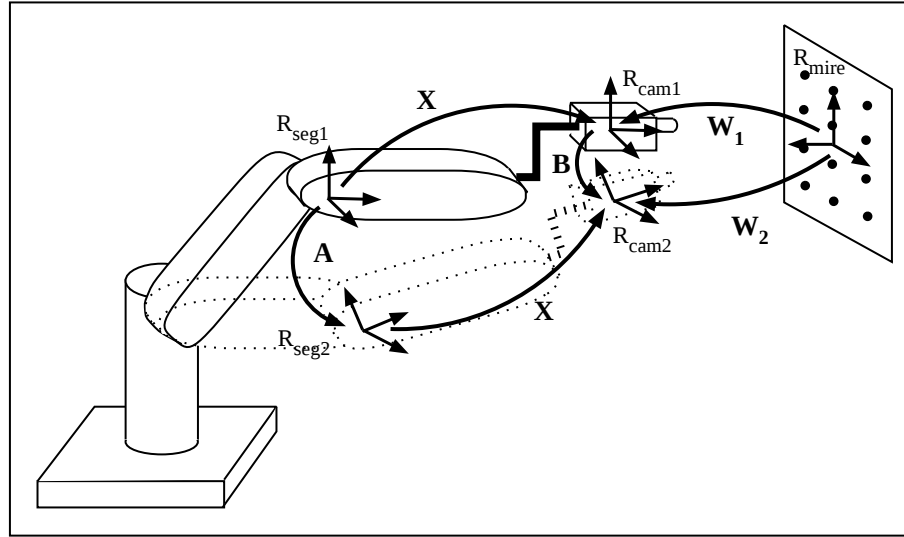
A est aisément connue au moyen du changeur de coordonnées : V_1 et V_2 sont déterminées à partir des coordonnées articulaires du robot en position 1 et 2 et $A = V_1^{-1}.V_2$ (voir figure 5.2).

5.1.3 Détermination de B

Dans la littérature, c'est souvent la position d'un capteur qui doit être déterminée. Pour connaître B , ce capteur, une caméra la plupart du temps, est calibré dans les deux positions par rapport à une mire fixe, ce qui permet de connaître W_1 et


 FIG. 5.2 - Le problème consiste à résoudre l'équation matricielle : $A.X = X.B$

W_2 et donc $B = W_1^{-1}.W_2$. La mire peut être constituée par exemple de points et de droites de contrôle placées à des positions connues dans l'espace [TL88] (voir figure 5.3).


 FIG. 5.3 - Détermination de B dans le cas d'une caméra portée par un robot

Dans notre cas, il fallait un système pour déterminer R_{marq1} et R_{marq2} par rapport à un référentiel fixe. Nous avons utilisé le système Optotrak (voir l'annexe A).

Dans la suite de ce chapitre, nous allons considérer que le marqueur à calibrer est matérialisé par un "rigid body" optotrak. Il sera ensuite facile de repérer notre véritable marqueur par rapport au "rigid body" au moyen d'un palpeur (voir photo

5.4).

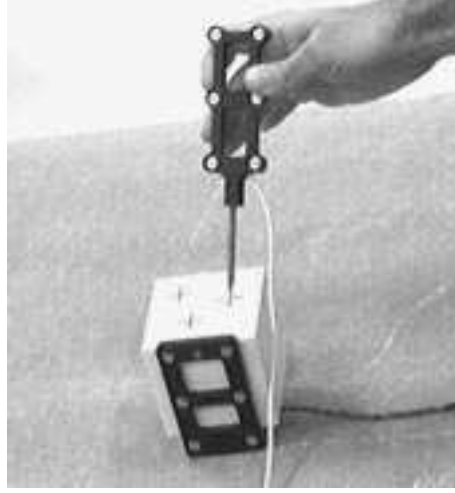


FIG. 5.4 - *On palpe les points caractéristiques du marqueur pour les repérer par rapport à la plaquette optotrak*

5.1.4 Résolution de $A.X = X.B$

A et B ayant été déterminées, il s'agit d'en déduire X . Notons qu'une équation de la forme $A.X = X.B$ possède un degré de liberté en rotation et un degré de liberté en translation [SA87]. Pour déterminer X de manière unique, il faut donc faire effectuer deux mouvements au bras du robot, ce qui permet de former le système :

$$\begin{cases} A_1.X = X.B_1 \\ A_2.X = X.B_2 \end{cases}$$

La méthode la plus couramment développée pour déterminer X consiste à décomposer A , B et X en rotation et translation, puis à résoudre consécutivement les deux systèmes engendrés :

$$\begin{cases} R_{A1}.R_X = R_X.R_{B1} \\ R_{A2}.R_X = R_X.R_{B2} \end{cases} \quad \text{puis} \quad \begin{cases} R_{A1}.T_X + T_{A1} = R_X.T_{B1} + T_X \\ R_{A2}.T_X + T_{A2} = R_X.T_{B2} + T_X \end{cases}$$

On peut résoudre ces deux systèmes de diverses manières. Nous en présentons deux, que nous avons implémentées et dont les résultats seront détaillés ultérieurement :

- Horaud [Hor] a explicité clairement une méthode exposée par Tsai [TL88]. Il préconise de représenter une rotation sous forme vectorielle. Le vecteur représente l'axe de la rotation et son module représente l'angle (voir [Aya89] pour une expression complète de cette décomposition à partir de la matrice de rotation).

Puis par une suite de raisonnements géométriques (les vecteurs de rotation des matrices A , B et X étant respectivement, $\vec{k}_A$, $\vec{k}_B$ et $\vec{k}_X$, on décompose $\vec{k}_A$ et $\vec{k}_B$ selon deux directions : l'une parallèle, l'autre perpendiculaire à $\vec{k}_X$...), on aboutit à un système linéaire dont la résolution permet d'évaluer l'axe ($\vec{k}_X$) et l'angle de la rotation R_X . Une fois R_X déterminée, il suffit de calculer T_X en résolvant aux moindres carrés le système :

$$\begin{cases} (R_{A1} - I_3) . T_X &= R_X . T_{B1} - T_{A1} \\ (R_{A2} - I_3) . T_X &= R_X . T_{B2} - T_{A2} \end{cases}$$

- Shiu et Ahmad [SA87] montrent qu'une solution générale de l'équation $R_A . R_X = R_X . R_B$ est : $R_X = Rot(\vec{k}_A, \beta) . R_{XP}$, où $\vec{k}_A$ est l'axe de rotation de R_A , R_{XP} est une solution particulière de l'équation $R_A . R_X = R_X . R_B$, et β est un angle arbitraire.

D'autre part, toute rotation qui satisfait $\vec{k}_A = R . \vec{k}_B$ est solution de $R_A . R_X = R_X . R_B$. Donc on peut choisir comme solution particulière $R_{XP} = Rot(\vec{v}, \omega)$ avec $\vec{v} = \vec{k}_B \wedge \vec{k}_A$ et $\omega = arctan2(|\vec{k}_B \wedge \vec{k}_A|, \vec{k}_B . \vec{k}_A)$.

Ayant une formulation générale pour la solution de chacune des deux équations $R_{A1} . R_X = R_X . R_{B1}$ et $R_{A2} . R_X = R_X . R_{B2}$, on cherche la solution du système qui doit vérifier : $Rot(\vec{k}_{A1}, \beta) . R_{XP1} = Rot(\vec{k}_{A2}, \beta) . R_{XP2}$. Ceci nous conduit à résoudre un système linéaire dont les inconnues sont $cos(\beta_1)$, $sin(\beta_1)$, $cos(\beta_2)$ et $sin(\beta_2)$. Comme le nombre d'équations est plus important que le nombre d'inconnues et qu'il y a du bruit pour des mesures réelles, on résout ce système aux moindres carrés.

Une fois R_X déterminée, on procède comme avec la méthode de Horaud pour calculer la translation.

On peut également se reporter à [Wan92] qui propose une méthode proche de celle de [Hor] et [TL88] ou à [CK91], [FH86] et [Hor87] qui décrivent trois autres manières de résoudre le problème en utilisant des quaternions unitaires pour représenter les rotations. Quant à Zhuang [ZS93] il explicite une méthode itérative permettant de déterminer simultanément la rotation et la translation, ce qui élimine la propagation d'erreurs.

L'inconvénient majeur de toutes ces méthodes est que leur champ d'application exclut le cas où les deux mouvements du robot ont des axes de rotation parallèles.

Lors d'une manipulation de calibrage classique, on peut éviter cette situation, en jouant sur les six degrés de liberté du robot.

Dans notre cas, le problème se pose pour le calibrage des marqueurs portés par le premier segment du robot, puisque l'on ne dispose alors que de la rotation de ce segment par rapport à la base. La difficulté est encore plus grande dans le cas où l'on veut calibrer un marqueur sur la base elle-même, puisque l'on ne dispose d'aucun mouvement.

Chapitre 5. Le calibrage des marqueurs

Il existe cependant d'autres méthodes de résolution, telle que celle décrite dans [Che91b], basée sur la représentation d'une transformation par un "écrou", qui, dans le pire des cas, détermine 5 des 6 paramètres. Un déplacement rigide peut être décrit de manière générale comme une rotation autour d'un axe et une translation le long de cet axe. C'est ce que l'on appelle un écrou. La méthode de [Che91b] considère donc le mouvement complet comme une seule entité géométrique (un axe et deux scalaires) et ne découple pas la rotation et la translation, ce qui permet un traitement unifié de tous les paramètres.

Cependant, cette méthode ne résout pas notre problème puisqu'elle ne permet de déterminer que 5 des 6 paramètres du changement de repères, lorsque le marqueur à calibrer est situé sur le premier segment du robot et ne fournit pas de solution pour les marqueurs situés sur la base. L'impossibilité de connaître complètement la position du marqueur lorsque l'on ne dispose que d'un seul degré de liberté, est illustrée par la figure 5.5 sur laquelle sont représentées deux rotations dont les axes sont confondus.

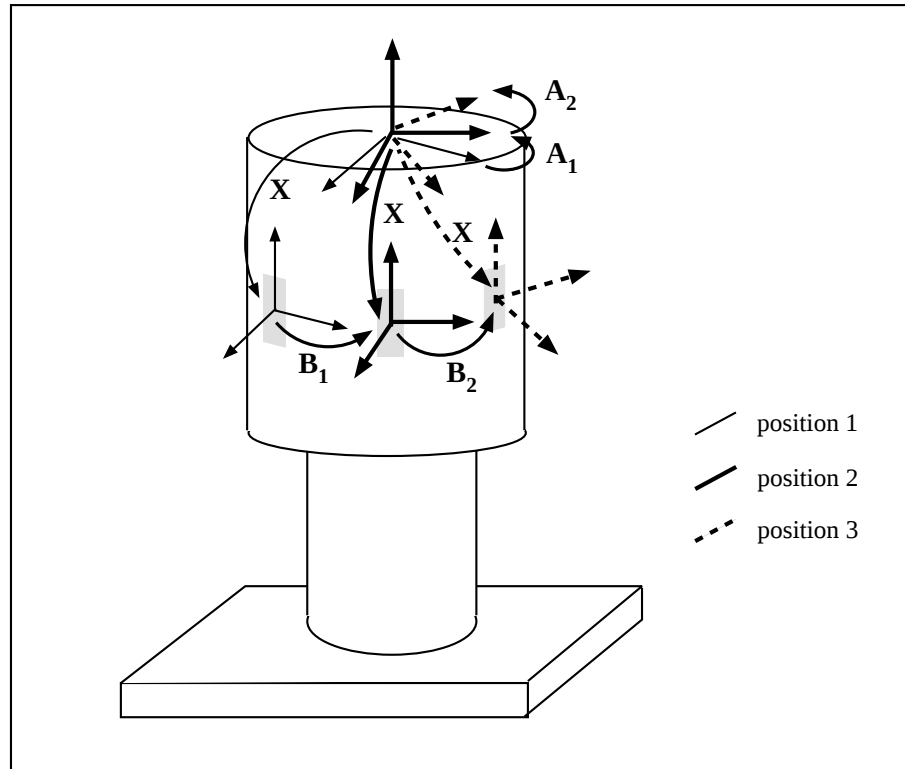


FIG. 5.5 - X ne peut pas être complètement déterminé lorsque les 2 rotations ont des axes parallèles

Sur cette figure, il est évident que si l'on fait effectuer (par exemple) une translation verticale au marqueur, les changements de repères A_1 , A_2 , B_1 et B_2 seront identiques, bien que X soit différent. Ceci montre clairement que l'un des paramètres

au moins ne peut pas être déterminé.

Ainsi les méthodes de résolution de " $A.X = X.B$ " nécessitent deux mouvements dont les axes des rotations ne sont pas parallèles, ce que l'on ne peut pas obtenir pour la base et le premier segment du robot, si l'on prend comme repère de référence le repère attaché à la base du robot (R_{robot}) ou un repère fixe par rapport à la base du robot. En revanche, on peut choisir comme repère de référence, un repère attaché à l'un des segments ultérieurs au deuxième segment pour calibrer les marqueurs de la base, et ultérieurs au troisième segment pour calibrer les marqueurs du premier segment. On pourra ainsi bénéficier des degrés de liberté des premières articulations et donc appliquer les méthodes classiques de résolution de " $A.X = X.B$ ".

Sur la figure 5.6, on voit comment interpréter un mouvement lorsque l'on prend R_{seg} comme repère de référence.

- On connaît aisément A par $A = V_1^{-1}.V_2$ où V_1 et V_2 sont calculées au moyen des matrices inverses du changeur de coordonnées.
- L'optotrak permet d'évaluer W_1 et W_2 ; on a recours à R_{marq} , afin d'estimer W_1 et W_2 par rapport à un référentiel fixe dans R_{seg} . On a alors de manière immédiate $B = W_1^{-1}.W_2$.

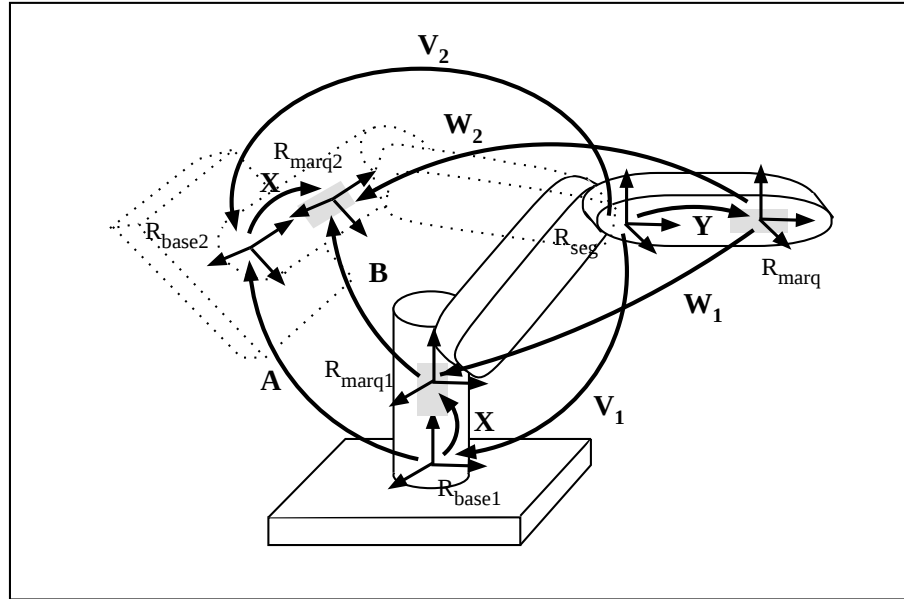


FIG. 5.6 - Détermination de la position d'un marqueur situé sur la base ou le premier segment du robot

Notons que l'on n'a pas besoin d'avoir recours à la résolution d'un système $A.X = X.B$, puisque l'on a de manière directe $X = V_1^{-1}.Y.W_1$. En effet la position et l'orientation de R_{marq} par rapport à R_{seg} (donnée par la matrice Y) peuvent

être calculées auparavant grâce à une phase de calibrage classique (résolution d'un système $A.Y = Y.B$) appliquée au couple de référence (R_{seg}, R_{marq}).

Cependant, cette manière de procéder introduirait les erreurs commises lors de la détermination de Y . Il est donc préférable de résoudre un système $A.X = X.B$, qui est totalement indépendant des résultats antérieurs.

5.2 Précision de la résolution

5.2.1 Précision de A

5.2.1.1 Répétabilité du positionnement du robot

La répétabilité est la manière dont le robot reproduit une tâche apprise. Nous l'avons évaluée pour chacune des articulations du robot. Pour chacune des articulations, nous avons effectué une série de mesures. Chaque mesure est caractérisée par le choix d'une configuration de référence. Le modus operandi d'une mesure est alors le suivant :

- on fixe un laser à l'extrémité du robot.
- on place le robot de telle sorte que :
 - le laser se projette à peu près perpendiculairement sur un mur (dans notre cas, la distance laser-mur D était d'environ 3 mètres).
 - l'axe de rotation de l'articulation à tester soit parallèle au mur afin d'amplifier au maximum les mesures : une petite déviation α se traduit en projection par une différence d d'autant plus amplifiée que la distance au mur D est grande (voir figure 5.7).

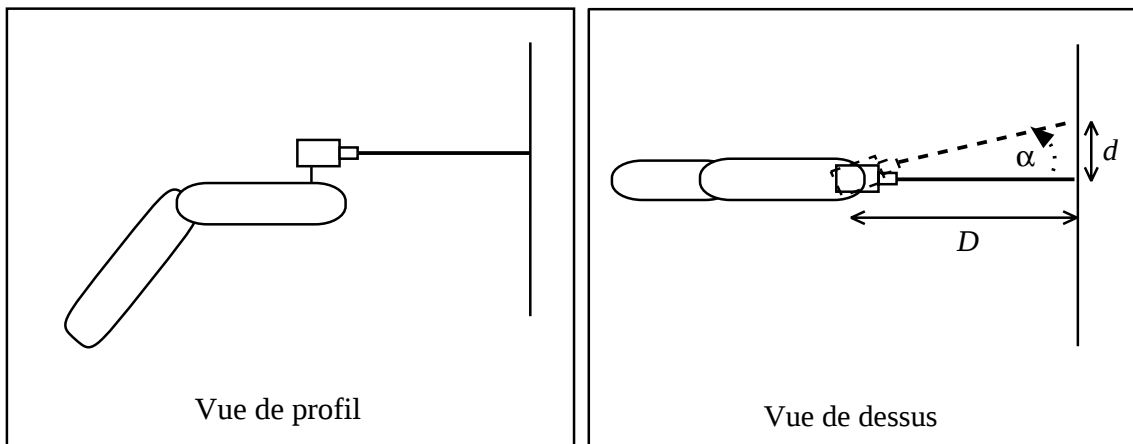


FIG. 5.7 - Mesure de la répétabilité dans le cas où l'axe de rotation est parallèle à la surface de projection du laser

Si l'axe de rotation était perpendiculaire au mur, on se trouverait dans la situation de la figure 5.8.

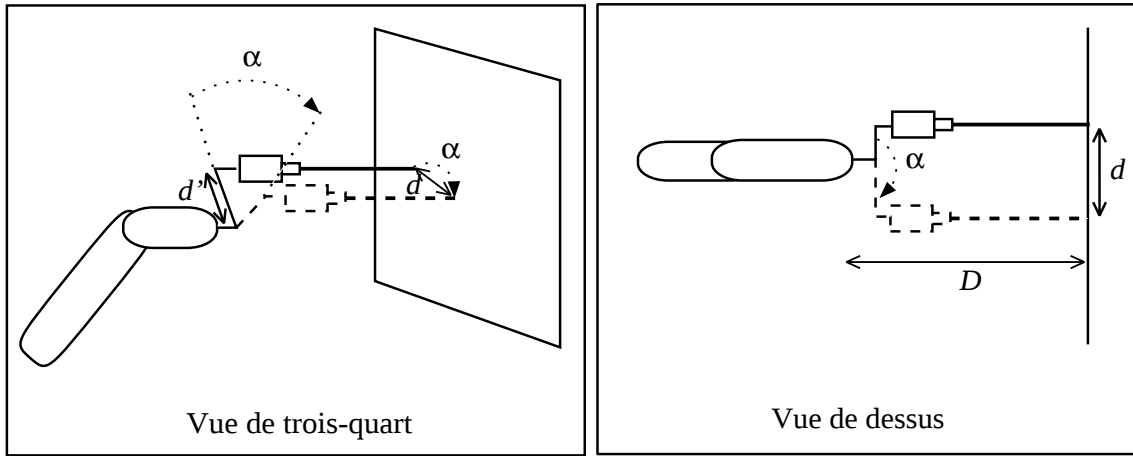


FIG. 5.8 - *Mesure de la répétabilité dans le cas où l'axe de rotation est perpendiculaire à la surface de projection du laser*

On voit que les différences seraient reportées telles quelles sur le mur de projection. En effet, D ne jouerait plus aucun rôle et seule interviendrait la distance d' entre le rayon laser et l'axe de rotation. Or $d' \ll D$. L'estimation de la déviation serait donc moins précise.

La configuration du robot constitue alors la configuration de référence.

- on relève la position de la tache du laser sur le mur, dans cette configuration de référence.
- pour une série de configurations différentes de la configuration de référence (ces configurations, au nombre de 10 lors de nos expérimentations, ne diffèrent que pour la composante dont on veut mesurer la répétabilité):
 - on fait déplacer le robot dans cette nouvelle configuration,
 - on commande au robot de revenir à la position de référence courante,
 - on relève la position de la tache du laser sur le mur et on mesure la distance entre la tache précédente et la tache de projection actuelle du laser.

Schématiquement, on est dans la situation de la figure 5.9 (la position précédente est repérée par l'angle α_1 , la position courante par l'angle α_2).

On mesure d sur le mur de projection. d est la distance entre les 2 taches de projection du laser. D est la distance entre le mur et la source d'émission du laser portée par le robot (dans notre cas 3 mètres: on néglige les petites variations selon cette direction, dues aux erreurs de position du robot, qui sont de l'ordre du

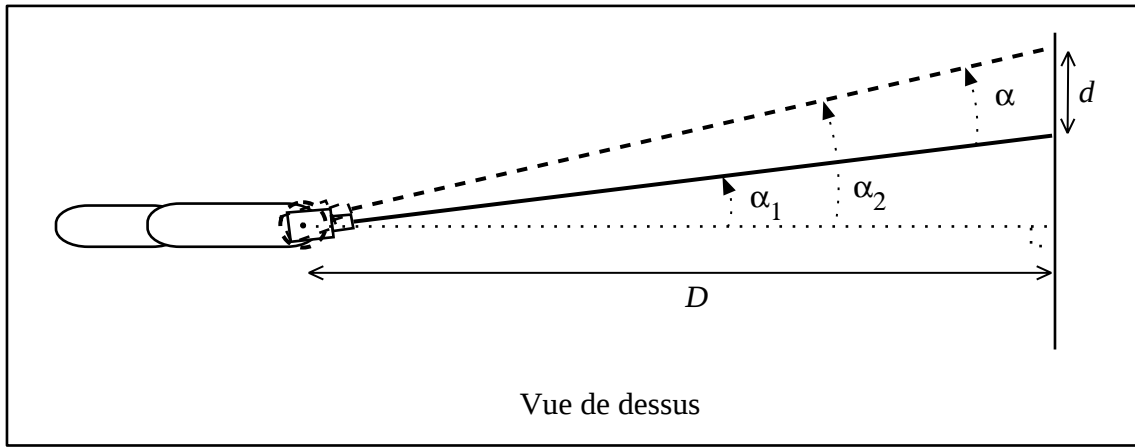


FIG. 5.9 -

millimètre, donc insignifiantes par rapport aux 3 mètres). La valeur exacte de d est :

$$d = D * (\tan(\alpha_2) - \tan(\alpha_1)) = D * (\tan(\alpha_1 + \alpha) - \tan(\alpha_1))$$

$$\text{D'où} \quad \alpha = \arctan\left(\frac{d}{D} + \tan(\alpha_1)\right) - \alpha_1$$

Par un développement du premier ordre, en considérant que :

$$\frac{d}{D} \ll 1$$

on obtient une valeur approchée de l'erreur angulaire :

$$\alpha = \arctan\left(\frac{d}{D}\right)$$

Les résultats obtenus sur le PUMA 260 du laboratoire TIMC sont répertoriés dans le tableau suivant :

	Ecart maximum entre plusieurs positions sans couper les moteurs	Ecart maximum entre plusieurs positions en coupant les moteurs	Ecart maximum entre la position d'apprentissage et toutes les autres positions
Axe 1	$d = 0.5 \text{ mm}$ $\alpha = 0.01^\circ$	$d = 2.0 \text{ mm}$ $\alpha = 0.04^\circ$	$d = 5.5 \text{ mm}$ $\alpha = 0.11^\circ$
Axe 2	$d = 0.5 \text{ mm}$ $\alpha = 0.01^\circ$	$d = 1.5 \text{ mm}$ $\alpha = 0.03^\circ$	$d = 5.5 \text{ mm}$ $\alpha = 0.11^\circ$
Axe 3	$d = 1.0 \text{ mm}$ $\alpha = 0.02^\circ$	$d = 1.5 \text{ mm}$ $\alpha = 0.03^\circ$	$d = 6.5 \text{ mm}$ $\alpha = 0.13^\circ$
Axe 4	$d = 1.0 \text{ mm}$ $\alpha = 0.02^\circ$	$d = 5.0 \text{ mm}$ $\alpha = 0.10^\circ$	$d = 18.0 \text{ mm}$ $\alpha = 0.35^\circ$
Axe 5	$d = 1.5 \text{ mm}$ $\alpha = 0.03^\circ$	$d = 1.5 \text{ mm}$ $\alpha = 0.03^\circ$	$d = 20.0 \text{ mm}$ $\alpha = 0.39^\circ$
Axe 6	$d = 5.0 \text{ mm}$ $\alpha = 0.10^\circ$	$d = 13.0 \text{ mm}$ $\alpha = 0.25^\circ$	$d = 22.0 \text{ mm}$ $\alpha = 0.43^\circ$

Les résultats de la première colonne sont les distances maximales entre deux taches obtenues au cours d'une même série de mesures (on indique également les angles correspondant à ces distances). A l'intérieur de cette série, on ne prend pas en compte la première position, obtenue en positionnant manuellement le robot. Cette première position, que nous dénommons "position d'apprentissage", est en effet particulière car elle ne bénéficie pas d'une correction des jeux. Pour les autres positions, la correction des jeux consiste simplement à effectuer les déplacements finaux toujours dans le même sens.

Les résultats de la deuxième colonne ont été obtenus de la même manière que ceux de la première colonne, à la différence près qu'entre chaque repérage de la tache de projection du laser, on a coupé les moteurs du robot.

Enfin, la troisième colonne fournit la distance maximale, ainsi que l'angle correspondant, entre la position d'apprentissage et toutes les autres positions obtenues lors des deux tests précédents.

Les tests décrits précédemment permettent d'évaluer la répétabilité du positionnement du robot selon les coordonnées articulaires. En vue d'estimer une majoration de la répétabilité globale, pour chaque axe, on a mesuré la distance maximale qui le sépare de l'extrémité du robot (axe 1 : 70 cm ; axe 2 : 70 cm ; axe 3 : 47 cm ; axe 4 : 20 cm ; axe 5 : 20 cm ; axe 6 : 5 cm). On a ensuite rapporté les erreurs du tableau précédent, obtenues pour une distance de 3 m, aux distance mesurées. La somme de ces erreurs nous fournit une majoration de l'erreur de répétabilité à l'extrémité du robot. Pour les valeurs obtenues en coupant les moteurs entre chaque série, ce qui

Chapitre 5. Le calibrage des marqueurs

correspond à une utilisation classique du robot, on a :

$$e_{max} = \frac{2.0 * 70 + 1.5 * 70 + 1.5 * 47 + 5.0 * 20 + 1.5 * 15 + 13.0 * 5}{300} \approx 1.7 \text{ mm}$$

Ces résultats constituent uniquement une mesure de répétabilité, et non pas une mesure de précision.

5.2.1.2 Précision du changeur de coordonnées

La précision du changeur de coordonnées n'a pas été évaluée, mais la procédure à suivre pourrait être la suivante :

- On fixe une configuration du robot, ce qui fixe une position (et orientation) du laser porté par le robot.
- On calcule, par le changeur inverse, les 8 configurations qui permettent d'amener l'effecteur du robot dans la position/orientation fixée.
- Si le modèle du robot était parfait, le laser se projetterait exactement au même endroit dans les 8 configurations. Comme lors de l'évaluation de la répétabilité, on mesure la distance entre les diverses taches lumineuses, ce qui nous fournit une indication sur la précision du changeur de coordonnées.

Cette manière de procéder inclura également les erreurs liées à la répétabilité.

Notons que l'on peut également se faire une bonne idée de la précision par l'intermédiaire des résidus du calibrage.

5.2.1.3 Précision de la transformation A

Soit $(P)_{R_{seg}}$ un point servant à caractériser le repère R_{seg} . Par exemple $P = (0 \ 0 \ 0)$ ou $P = (1 \ 0 \ 0)$ ou $P = (0 \ 1 \ 0)$ ou $P = (0 \ 0 \ 1)$. Dans le repère R_{robot} , les coordonnées de ce point sont calculées par : $(P)_{R_{robot}} = V_i.(P)_{R_{seg}}$ dans la position i (voir figure 5.2). L'erreur maximale qui peut être commise sur les coordonnées de P par ce calcul, matérialise la précision du changeur. La précision de l'inverse de V_i est la même que celle de V_i puisqu'il s'agit d'une matrice de passage. L'erreur engendrée par l'application successive de deux de ces matrices est majorée par deux fois l'erreur engendrée par l'application de l'une d'elles.

Donc l'erreur commise sur A est majorée par deux fois l'erreur commise par le changeur du robot.

5.2.2 Précision de B

5.2.2.1 Répétabilité de l'Optotrak

On s'est contenté de placer un "rigid body" dans le champ des caméras et de saisir plusieurs fois sa position estimée par Optotrak. Les mesures varient de 0.04 mm selon les directions x et y , et de 0.06 mm selon l'axe z , c'est-à-dire selon un axe parallèle à l'axe optique de la caméra centrale de l'Optotrak.

5.2.2.2 Précision de l'Optotrak

La DER de l'EDF a fait effectuer des mesures de précision de son système optotrak. Les mesures de références ont été réalisées par un interféromètre Hewlett-Packard de type 5518 A. Les résultats obtenus pour un volume d'utilisation légèrement supérieur au nôtre (x et y compris entre -0.6 m et 0.6 m et z aux environs de 3 m) sont les suivants : incertitude en x : 0.44 mm ; incertitude en y : 0.20 mm ; incertitude en z : 0.70 mm .

On peut également se reporter aux mesures que nous avons décrites au paragraphe 3.2.3.2, qui correspondent plus exactement au volume que nous utilisons.

5.2.2.3 Précision de la transformation B

Par un raisonnement similaire à celui concernant la précision de A , on déduit que les erreurs de précision de B sont majorées par deux fois les erreurs de précision de l'Optotrak.

5.2.3 Précision de la solution obtenue par les deux méthodes étudiées

Nous avons implémenté deux méthodes pour résoudre le problème $A.X = X.B$: celle de Horaud [Hor] et celle de Shiu et Ahmad [SA87]. Leur principe est exposé au paragraphe 5.1.4. Pour chacune d'elles, nous donnons des résultats obtenus à partir de données simulées exactes, puis nous évaluons ces méthodes sur des données simulées bruitées. Cette évaluation se fait en comparant une matrice X_{ref} que l'on s'est imposée et une matrice X_{cal} , retrouvée par le calcul. Nous commençons par présenter la méthode de comparaison.

5.2.3.1 Avant-propos : comparaison de deux matrices

La première idée est de calculer la différence entre les matrices : $X_{ref} - X_{cal}$. Mais analyser cette différence n'est pas aisé de manière immédiate puisque l'on doit intégrer des erreurs de translation et de rotation. Quel poids accorder à chaque type d'erreur ?

Un moyen simple d'homogénéiser toutes les valeurs consiste à appliquer les matrices X_{ref} et X_{cal} à des points, puis à comparer les points ainsi transformés. On a choisi un ensemble de 8 points (P_i , $i = 1..8$) disposés sur les coins d'un cube centré sur l'origine. Notons que choisir la distance des points à l'origine revient à choisir une pondération des erreurs de translation et de rotation (une distance faible met en avant les erreurs de translation, alors qu'une distance importante met l'accent sur les erreurs de rotation). On applique X_{ref} et X_{cal} aux P_i et l'on obtient $P_{i,ref}$ et $P_{i,cal}$.

Si X_{ref} et X_{cal} étaient parfaitement identiques, pour tout i , $P_{i,ref}$ et $P_{i,cal}$ seraient confondus. Donc la distance entre ces points nous fournit une évaluation de

Chapitre 5. Le calibrage des marqueurs

la ressemblance de X_{ref} et de X_{cal} . Ainsi, une fonction de comparaison peut être :

$$Fcomp(X_{ref}, X_{cal}) = \sum_{i=1}^8 d(P_{i,ref}, P_{i,cal}) = \sum_{i=1}^8 d(X_{ref}P_i, X_{cal}P_i)$$

On normalise cette fonction en divisant par le nombre de points :

$$Fcomp_{moy}(X_{ref}, X_{cal}) = \frac{1}{8} Fcomp(X_{ref}, X_{cal})$$

On peut également choisir pour fonction de comparaison la distance maximale entre deux points correspondants $P_{i,ref}$ et $P_{i,cal}$:

$$Fcomp_{max}(X_{ref}, X_{cal}) = \max_{i=1}^8 d(P_{i,ref}, P_{i,cal}) = \max_{i=1}^8 d(X_{ref}P_i, X_{cal}P_i)$$

Par la suite, nous appliquons ce calcul avec 8 points situés sur un cube de 120 mm de côté (les points sont donc à une distance $d \approx 104$ mm de l'origine). Cette distance correspond à notre champ d'utilisation. En effet, la translation entre le repère associé au segment (R_{seg}) et le repère associé au marqueur (R_{marq}) est limitée par la longueur des segments du PUMA, longueur approximée à une dizaine de centimètres.

5.2.3.2 Méthode de R.Horaud

Résultats de simulation

Ces résultats ont été obtenus en trois étapes :

- Le processus pour créer des données simulées consiste à fixer B_1 , B_2 et X_{ref} , puis à calculer $A_1 = X_{ref}.B_1.X_{ref}^{-1}$ et $A_2 = X_{ref}.B_2.X_{ref}^{-1}$.
- On applique ensuite la méthode de Horaud pour trouver X_{cal} solution du système d'équations :

$$\begin{cases} A_1.X &= X.B_1 \\ A_2.X &= X.B_2 \end{cases}$$

- Enfin, on compare X_{ref} (imposé initialement) et X_{cal} .

Nous avons simulé plusieurs situations pour B_1 et B_2 :

- B_1 et B_2 ont des axes de rotation très différents. Alors nous avons envisagé diverses valeurs d'angle de rotation pour B_1 et B_2 :
 - angle proche de 0;

- angle égal à π ;
- angle égal à $\pi/2$;
- angle "quelconque".
- B_1 et B_2 ont des axes de rotation parallèles.
- B_1 et B_2 ont des axes de rotation presque parallèles. On envisage les mêmes possibilités pour les angles de rotation que dans le cas où les axes de rotation sont très différents.

La nature de X peut également influencer la précision des résultats. La méthode exposée dans [Wan92] se révèle être meilleure lorsque l'angle de la rotation de X est proche de $\frac{\pi}{2}$. Notons qu'on peut se ramener à cette situation en utilisant un repère intermédiaire décalé de $\frac{\pi}{2}$ par rapport à une estimation a priori de X .

Pour nos simulations, nous avons utilisé un X_{ref} a priori quelconque (angle de rotation valant 60°), si ce n'est que nous avons choisi une translation de norme valant approximativement 170 mm , ce qui est légèrement supérieur aux conditions dans lesquelles nous nous trouvons avec le PUMA.

Lorsque les hypothèses ne sont pas respectées, c'est-à-dire lorsque les deux mouvements du robot ont des axes de rotation parallèles, les résultats obtenus en simulation sont très mauvais. La méthode de Horaud fournit une rotation avec des erreurs de l'ordre des valeurs suivantes : l'axe de rotation dévie de 1.5° ; l'erreur de l'angle de rotation est de l'ordre de 10^{-4}° ; quant à la translation fournie par la méthode de Horaud, elle est complètement fausse.

Lorsque B_1 et B_2 ont des axes de rotation presque parallèles (différence d'orientation égale successivement à 1° , 3° et 10°), on obtient les résultats suivants. Nous distinguons le cas où l'angle de l'une des rotations (R_{B_1} ou R_{B_2}) est petit (angle inférieur à 5°) du cas plus général :

	différence d'orientation des vecteurs rotation de B_1 et B_2	erreur d'orientation du vecteur de rotation	erreur de l'angle de rotation	erreur en translation
cas où l'angle de l'une des rotations est $< 5^\circ$	1°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	0.038 %
	3°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	10^{-3} %
	10°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	$< 10^{-3}$ %
cas général	1°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	0.003 %
	3°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	$< 10^{-3}$ %
	10°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	$< 10^{-3}$ %

Chapitre 5. Le calibrage des marqueurs

On remarque donc que pour des données exactes, les résultats sont très bons, même si les conditions (axes des rotations presque parallèles ou rotations presque nulles) sont proches des cas pour lesquels la méthode ne peut pas fournir de résultats.

Enfin, dans le cas général où les axes de rotation de B_1 et B_2 sont très différents, les résultats obtenus sont encore meilleurs : différence d'orientation des vecteurs rotation et différence des angles de rotation inférieures à 10^{-5}° ; erreur en translation inférieure à $10^{-4} \%$.

Résultats de simulation avec du bruit

TAB. 5.1 - Résultats de la fonction $Fcomp_{moy}$ sur des données simulées bruitées. XXX indique une erreur trop importante ($Fcomp_{moy} > 10.0 \text{ mm}$).

	différence d'orientation des vecteurs rotation de B_1 et B_2	$Fcomp_{moy}$ (X_{ref}, X_{cal}) avec $BruitMax$ $= 0.10$	$Fcomp_{moy}$ (X_{ref}, X_{cal}) avec $BruitMax$ $= 0.15$	$Fcomp_{moy}$ (X_{ref}, X_{cal}) avec $BruitMax$ $= 0.25$	$Fcomp_{moy}$ (X_{ref}, X_{cal}) avec $BruitMax$ $= 0.50$
cas où l'angle de l'une des rotations est $< 5^\circ$	5°	XXX	XXX	XXX	XXX
	10°	XXX	XXX	XXX	XXX
	15°	XXX	XXX	XXX	XXX
	20°	XXX	XXX	XXX	XXX
	25°	XXX	XXX	XXX	XXX
	$> 30^\circ$	XXX	XXX	XXX	XXX
cas général	5°	XXX	XXX	XXX	XXX
	10°	4.82	6.92	XXX	XXX
	15°	4.24	6.18	XXX	XXX
	20°	2.74	5.09	8.54	XXX
	25°	2.18	2.81	5.48	XXX
	$> 30^\circ$	1.97	2.71	4.97	XXX

Le processus est le suivant. On matérialise un marqueur par les trois points P_0 , P_1 et P_2 dont les coordonnées sont exprimées en millimètres dans le repère R_{marq} : $P_0 = (0 \ 0 \ 0)$, $P_1 = (50 \ 0 \ 0)$ et $P_2 = (0 \ 50 \ 0)$.

- On crée des données simulées non bruitées en fixant B_1, B_2 et X_{ref} , puis en calculant $A_1 = X_{ref} \cdot B_1 \cdot X_{ref}^{-1}$ et $A_2 = X_{ref} \cdot B_2 \cdot X_{ref}^{-1}$.
- On transforme P_i ($i = 0..2$) par B_1 pour obtenir P'_i ($i = 0..2$).

TAB. 5.2 - Résultats de la fonction $Fcomp_{moy}$ sur des données simulées bruitées (XXX indique une erreur trop importante ($Fcomp_{max} > 10.0$ mm)).

	différence d'orientation des vecteurs rotation de B_1 et B_2	$Fcomp_{max}$ (X_{ref}, X_{cal}) avec $BruitMax$ = 0.10	$Fcomp_{max}$ (X_{ref}, X_{cal}) avec $BruitMax$ = 0.15	$Fcomp_{max}$ (X_{ref}, X_{cal}) avec $BruitMax$ = 0.25	$Fcomp_{max}$ (X_{ref}, X_{cal}) avec $BruitMax$ = 0.50
cas où l'angle de l'une des rotations est $< 5^\circ$	5°	XXX	XXX	XXX	XXX
	10°	XXX	XXX	XXX	XXX
	15°	XXX	XXX	XXX	XXX
	20°	XXX	XXX	XXX	XXX
	25°	XXX	XXX	XXX	XXX
	$> 30^\circ$	XXX	XXX	XXX	XXX
cas général	5°	XXX	XXX	XXX	XXX
	10°	4.99	7.23	XXX	XXX
	15°	4.57	6.82	XXX	XXX
	20°	3.17	5.92	9.93	XXX
	25°	2.33	3.31	6.33	XXX
	$> 30^\circ$	2.08	3.21	5.34	XXX

- On bruite les points P'_i pour obtenir P''_i ($i = 0..2$). Pour chacune des coordonnées :
 - on génère une variable aléatoire $Alea$ comprise entre 0 et 1.
 - on ajoute à la coordonnée concernée un facteur $2*BruitMax*(Alea-0.5)$.
- On reconstruit une matrice B'_1 à partir des P''_i :
 - le vecteur translation de la matrice est P''_0 .
 - la première colonne de la matrice est le vecteur :

$$\vec{x} = \frac{P''_0 \vec{P''_1}}{\|P''_0 \vec{P''_1}\|}$$

- la deuxième colonne est un vecteur perpendiculaire à $\vec{x}$ et appartenant au plan défini par $\vec{x}$ et $P''_0 \vec{P''_2}$:

$$\vec{y} = \frac{P''_0 \vec{P''_2} - (\vec{x} \cdot P''_0 \vec{P''_2}) \vec{x}}{\|P''_0 \vec{P''_2} - (\vec{x} \cdot P''_0 \vec{P''_2}) \vec{x}\|}$$

- la troisième colonne est le vecteur : $\vec{z} = \vec{x} \wedge \vec{y}$

Chapitre 5. Le calibrage des marqueurs

- On procède de même pour B_2, A_1 et A_2 . On obtient ainsi des matrices bruitées B'_2, A'_1 et A'_2 . Nous avons utilisé la même valeur de $BruitMax$ pour toutes les matrices. On aurait pu affiner les simulation en choisissant des valeurs différentes pour les matrices A et les matrices B , ce qui aurait traduit les différentes précisions qu'il est possible d'obtenir pour le robot et l'optotrak.
- On applique ensuite la méthode de Horaud pour trouver X_{cal} solution du système d'équations :

$$\begin{cases} A'_1.X &= X.B'_1 \\ A'_2.X &= X.B'_2 \end{cases}$$

- Enfin, on compare X_{ref} , imposé initialement, et X_{cal} au moyen des fonctions décrites en 5.2.3.1.

Les résultats obtenus sont fournis dans les tableaux 5.1 (pour $Fcomp_{moy}$) et 5.2 (pour $Fcomp_{max}$).

Il est évident que lorsque l'une des rotations est trop petite, la méthode ne supporte pas le bruit. Il en est de même lorsque les vecteurs rotation des deux rotations de B_1 et B_2 sont proches l'un de l'autre. Nous ne pourrions donc utiliser la méthode de Horaud que si les conditions suivantes sont remplies :

- B_1 et B_2 ont des rotations d'angle supérieur à 10° .
- les vecteurs de rotation de B_1 et de B_2 forment un angle supérieur à 30° .

Signalons également que lorsque l'une des rotations est égale à π (ou proche de π), il arrive que la méthode de Horaud fournisse des résultats complètement faux.

On peut comparer ces résultats à ceux exposés dans [TL88] dont la méthode de résolution est très similaire à celle de Horaud. Dans cet article, Tsai et Lenz constatent que l'erreur dans la détermination de la rotation est inversement proportionnelle d'une part au sinus de l'angle formé par les axes des rotations A_1 et A_2 , d'autre part aux angles des rotations A_1 et A_2 .

Notons que 0.25 mm est un bruit assez réaliste et que les mesures que nous avons effectuées ont été obtenues en simulant des points situés sur un carré de 5 cm de côté. Elles seraient plus mauvaises avec un carré plus petit (amplification des erreurs dans la détermination de la rotation). Ceci signifie que nous devons choisir des marqueurs suffisamment grands pour être capable de les calibrer correctement avec cette méthode.

5.2.3.3 Méthode de Shiu

Résultats de simulation

On procède exactement de la même manière que pour la méthode de Horaud, en appliquant la méthode de Shiu. Les résultats obtenus sont les suivants :

- lorsqu'on ne respecte pas les hypothèses, la méthode de Shiu ne donne aucun résultat cohérent, alors que la méthode de Horaud fournit une rotation approximative.
- lorsque l'une des rotations est petite (angle $< 5^\circ$), les résultats obtenus par la méthode de Shiu sont moins bons : erreur de 0.03° dans l'orientation des vecteurs de rotation et erreur de 0.008% en translation (à titre de comparaison, la méthode de Horaud produit, dans les mêmes conditions, une erreur inférieure à 10^{-6}° dans l'orientation des vecteurs de rotation et une erreur de 0.001% en translation).
- cas où les axes de rotation sont presque parallèles :

	différence d'orientation des vecteurs rotation de B_1 et B_2	erreur d'orientation du vecteur de rotation	erreur de l'angle de rotation	erreur en translation
cas où l'angle de l'une des rotations est $< 5^\circ$	1°	0.093°	10^{-5}°	0.058%
	3°	0.034°	$< 10^{-5}^\circ$	0.007%
	10°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	$< 10^{-3}\%$
cas général	1°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	$10^{-3}\%$
	3°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	$< 10^{-3}\%$
	10°	$< 10^{-5}^\circ$	$< 10^{-5}^\circ$	$< 10^{-3}\%$

- dans tous les autres cas, les résultats sont excellents et tout-à-fait similaires à ceux obtenus par la méthode de Horaud : les erreurs d'orientation des vecteurs rotation et les erreurs des angles de rotation sont inférieures à 10^{-5}° ; les erreurs de translation sont inférieures à $10^{-4}\%$.

Résultats de simulation avec du bruit

On procède exactement de la même manière que pour la méthode de Horaud, en appliquant la méthode de Shiu. Les résultats obtenus sont tout-à-fait similaires, exceptions faites de deux points particuliers :

- Lorsque l'une des rotations est égale à π (ou proche de π), la méthode de Shiu fournit des résultats complètement erronés beaucoup plus fréquemment que la méthode de Horaud (une fois sur deux).

Chapitre 5. Le calibrage des marqueurs

- La valeur de $Fcomp_{max}$ est en général légèrement supérieure (différence d'environ 0.2 mm).

5.2.3.4 Conclusion

Au vu des résultats, la méthode de Horaud est meilleure lorsque les conditions sont difficiles : rotations trop faibles (inférieures à 10°), ou inclinaison entre les axes de rotation de A_1 et A_2 (respectivement B_1 et B_2) trop petite (inférieure à 30°). La différence est d'autant plus flagrante que le bruit est important.

En revanche, les méthodes de Horaud et de Shiu sont tout-à-fait équivalentes lorsqu'elles sont appliquées dans de bonnes conditions. Nous avons comparé les résultats obtenus par chacune des méthodes sur les mêmes données, à savoir une trentaine de (A_1, A_2, B_1, B_2) répondant au critère de "bonnes conditions". En moyenne, les différences entre X_{Horaud} et X_{Shiu} sont de l'ordre de 0.2 mm en translation, 0.05° dans l'orientation du vecteur de rotation et 0.03° dans l'angle de rotation. Les résultats sont donc assez similaires.

5.3 Stratégie globale

5.3.1 Fonction d'erreur

La fonction d'erreur doit nous permettre d'évaluer dans quelle mesure la matrice X obtenue est pertinente par rapport aux données initiales.

On veut donc comparer les matrices $A_i.X$ et $X.B_i$ pour $i (i = 1..2)$. Pour cela, nous procédons comme au paragraphe 5.2.3.1 en appliquant les matrices $A_1.X$ et $X.B_1$ (respectivement $A_2.X$ et $X.B_2$) à des points P_i , $i = 1..8$, puis en comparant les points ainsi transformés : P'_{1i} et P''_{1i} (respectivement P'_{2i} et P''_{2i}).

Si X était une solution parfaite, pour tout i , pour $j = \{1, 2\}$, P'_{ji} et P''_{ji} seraient confondus. Donc la distance entre ces points nous fournit une évaluation de la validité de X . Ainsi, une fonction d'évaluation de l'erreur peut être :

$$\begin{aligned} Ferr(X) &= \sum_{i=1}^8 (d(P'_{1i}, P''_{1i}) + d(P'_{2i}, P''_{2i})) \\ &= \sum_{i=1}^8 (d(A_1 X P_i, X B_1 P_i) + d(A_2 X P_i, X B_2 P_i)) \\ &= Fcomp(A_1 X, X B_1) + Fcomp(A_2 X, X B_2) \end{aligned}$$

On normalise cette fonction en divisant par le nombre de points :

$$Ferr_{moy}(X) = \frac{1}{16} Ferr(X) = \frac{1}{2} (Fcomp_{moy}(A_1 X, X B_1) + Fcomp_{moy}(A_2 X, X B_2))$$

On peut également choisir pour fonction d'évaluation la distance maximale entre deux points correspondants P'_{ji} et P''_{ji} :

$$Ferr_{max}(X) = \max_{i=1..8, j=1..2} d(P'_{ji}, P''_{ji})$$

$$\begin{aligned}
&= \max_{i=1..8, j=1..2} d(A_j X P_i, X B_j P_i) \\
&= \max(Fcomp_{max}(A_1 X, X B_1), Fcomp_{max}(A_2 X, X B_2))
\end{aligned}$$

Nous avons évalué cette fonction d'erreur sur une série de 32 mesures réelles. Chaque mesure est constituée de 2 matrices A (A_1 et A_2), de 2 matrices B (B_1 et B_2) et des matrices $X_{Horausd}$ et X_{Shiu} , résultats des systèmes d'équations

$$\begin{cases} A_1.X &= X.B_1 \\ A_2.X &= X.B_2 \end{cases}$$

par les méthodes de Horaud et de Shiu. Nous avons appliqué ce calcul avec 8 points situés sur un cube de 120 mm de côté. Les points sont donc à une distance $d \approx 104$ mm de l'origine, ce qui correspond à notre champ d'utilisation, étant donné que la longueur des segments du PUMA 260 est de cet ordre de grandeur.

Parmi cet ensemble de 32 résultats, nous ne retenons que les valeurs extrêmes, la valeur moyenne et l'écart-type.

	Valeur minimale	Valeur maximale	Valeur moyenne	Ecart type
$Ferr_{moy}(X_{Horausd})$	0.13	0.54	0.26	0.09
$Ferr_{max}(X_{Horausd})$	0.18	0.75	0.42	0.15
$Ferr_{moy}(X_{Shiu})$	0.06	0.37	0.22	0.08
$Ferr_{max}(X_{Shiu})$	0.09	0.71	0.37	0.15

Les fonctions d'erreur sont dans l'ensemble plus faibles pour des résultats obtenus par la méthode de Shiu que pour des résultats obtenus par la méthode de Horaud. Cette contradiction apparente avec nos conclusions du paragraphe 5.2.3.4 s'explique par le fait que les données sont bruitées. Par rapport à la méthode de Horaud, la méthode de Shiu peut donc davantage minimiser les erreurs sur des données bruitées, tout en fournissant un résultat plus éloigné de la matrice de référence X_{ref} non bruitée choisie au paragraphe 5.2.3.

5.3.2 Minimisation de la fonction d'erreur

On souhaite bien entendu obtenir la valeur d'erreur la plus petite possible. Pour cela, on modifie les paramètres de la matrice X (vecteur translation, vecteur de rotation pondéré par l'angle de rotation) par la méthode des gradients conjugués [PFTV92] qui est une fonction classique de minimisation : l'espace de recherche est un espace à 6 dimensions : 3 pour la translation, 3 pour la rotation. On peut se reporter au chapitre 1 de [Dor95] pour une excellente synthèse des différentes manières de représenter une rotation ou une transformation.

Le tableau suivant présente les résultats obtenus sur les mêmes données que précédemment, après une minimisation de la fonction $Ferr_{moy}$. On constate que $Ferr_{max}$ a également tendance à diminuer, ce qui confirme que la modification des paramètres de la transformation X se fait dans le bon sens :

	Valeur minimale	Valeur maximale	Valeur moyenne	Ecart type
$Ferr_{moy}(X_{H\acute{e}raud})$	0.09	0.41	0.19	0.06
$Ferr_{max}(X_{H\acute{e}raud})$	0.14	0.79	0.34	0.10
$Ferr_{moy}(X_{Shiu})$	0.05	0.29	0.17	0.06
$Ferr_{max}(X_{Shiu})$	0.08	0.62	0.30	0.10

5.3.3 Minimisation globale sur un ensemble de matrices

Jusqu'à présent, nous avons utilisé le minimum de données requises pour la résolution de $A.X = X.B$ (à savoir 2 matrices A_i et 2 matrices B_i). Ces données n'étant pas rigoureusement exactes, elles engendrent des erreurs dans la détermination de X . L'idée est donc d'utiliser un plus grand nombre d'informations afin d'atténuer l'influence du bruit. Ainsi, au lieu de minimiser une fonction d'erreur calculée sur 2 couples de matrices (A_i, B_i) , on minimise une fonction d'erreur calculée sur n couples (A_i, B_i) .

Cette nouvelle fonction d'erreur est définie de la manière suivante :

$$Fglob_{moy}(X) = \frac{1}{n} \sum_{i=1}^n Fcomp_{moy}(A_i.X, X.B_i)$$

La nouvelle fonction maximale serait :

$$Fglob_{max}(X) = \max_{i=1}^n Fcomp_{max}(A_i.X, X.B_i)$$

La valeur initiale de X est déterminée par l'une des méthodes de Horaud ou de Shiu appliquée à 2 des matrices, puis la minimisation s'effectue sur l'ensemble des matrices.

5.3.4 Algorithme

A partir des résultats de simulation du paragraphe 5.2.3.2 et de l'idée de minimisation globale du paragraphe 5.3.3, nous définissons la stratégie globale pour calibrer la position des marqueurs sur chaque segment du robot.

On commence par éliminer les couples (A_i, B_i) qui ne correspondent pas aux critères de "bonnes conditions" déterminés lors des phases de simulation. Puis l'on minimise la fonction d'erreur sur l'ensemble des couples restants, à partir de plusieurs valeurs initiales.

Notons que Zhuang et Shiu [ZS93] suggèrent également de procéder à une sélection des données à partir d'une "mesure d'observabilité".

Les étapes de la stratégie adoptée sont les suivantes :

1. Acquérir n couples de matrices (A_i, B_i) .

2. Vérifier que les angles de rotation de A_i et de B_i ne sont pas trop différents (différence inférieure à 2°). En effet, [SA87] montre que ces angles doivent être égaux puisque pour tout R_X , on a $R_A.R_X = R_X.R_B$. Eliminer le couple si tel n'est pas le cas.
3. Vérifier que les angles de rotation de A_i et de B_i sont inclus dans $[-170^\circ, -10^\circ] \cup [10^\circ, 170^\circ]$ (en effet, on a vu lors des simulations que les résultats n'étaient pas bons pour des angles proches de 0° ou 180°). Eliminer le couple si tel n'est pas le cas.
4. Regrouper par deux les couples restants de telle sorte que les vecteurs rotation de A_1 et A_2 (respectivement B_1 et B_2) forment un angle supérieur à 30° (l'angle doit être inclus dans $[30^\circ, 150^\circ]$).
5. Pour chaque groupe de deux couples fourni par l'étape précédente, appliquer la méthode de Horaud ou de Shiu afin d'obtenir une série de valeurs initiales de la matrice X .
6. Minimiser la fonction d'erreur globale à partir de chacune de ces valeurs initiales pour l'ensemble des matrices A_i et B_i non éliminées aux étapes 2 et 3, pour obtenir des matrices X optimisées.
7. Sélectionner la matrice X optimale, à savoir celle dont la fonction d'erreur globale est la plus faible.

5.4 Résultats

5.4.1 Données simulées bruitées

La stratégie définie au paragraphe précédent a été appliquée sur des données simulées bruitées afin de connaître quelle précision on peut en attendre. Ces données ont été obtenues de la manière suivante :

- génération aléatoire d'une matrice X_{ref} .
- génération aléatoire de 60 matrices B_i ($i = 1..60$).
- calcul des 60 matrices A_i ($i = 1..60$) correspondantes : $A_i = X_{ref}.B_i.X_{ref}^{-1}$.
- génération des matrices bruitées A'_i et B'_i ($i = 1..60$) à partir des A_i et B_i de la même manière qu'au paragraphe 5.2.3.2.

Nous avons donc calculé une matrice X_{cal} (que nous appellerons X_{Horaud} ou X_{Shiu} selon la méthode utilisée pour initialiser la minimisation) à partir de ces 60 paires de matrices (A'_i, B'_i) et nous avons obtenu les résultats suivants :

Chapitre 5. Le calibrage des marqueurs

<i>BruitMax</i> (en <i>mm</i>)		0.1	0.5	1.0
Erreur globale de la matrice de référence X_{ref} (erreur calculée avec les A'_i et B'_i bruitées ... bien sûr)	$Fglob_{moy}(X_{ref})$	0.37	2.05	3.88
	$Fglob_{max}(X_{ref})$	1.25	6.16	12.37
Erreur globale de la matrice calculée $X_{cal} = X_{Horausud}$ (initialisée par la méthode de Horaud)	$Fglob_{moy}(X_{cal})$	0.37	1.97	3.69
	$Fglob_{max}(X_{cal})$	1.16	6.16	12.55
Erreur de la matrice $X_{cal} = X_{Horausud}$ par rapport à la matrice de référence X_{ref}	$Fcomp_{moy}(X_{ref}, X_{cal})$	0.05	0.33	0.75
	$Fcomp_{max}(X_{ref}, X_{cal})$	0.08	0.44	1.03
Erreur globale de la matrice calculée $X_{cal} = X_{Shiu}$ (initialisée par la méthode de Shiu)	$Fglob_{moy}(X_{cal})$	0.37	1.98	3.69
	$Fglob_{max}(X_{cal})$	1.15	6.16	12.65
Erreur de la matrice $X_{cal} = X_{Shiu}$ par rapport à la matrice de référence X_{ref}	$Fcomp_{moy}(X_{ref}, X_{cal})$	0.07	0.33	0.76
	$Fcomp_{max}(X_{ref}, X_{cal})$	0.08	0.46	1.13

Il est clair que les résultats sont bien meilleurs qu'avec seulement 2 couples de matrices (A_i, B_i) (cf paragraphes 5.3.2). Ainsi, avec un bruit de 0.1 *mm*, on obtient $Fcomp_{moy}(X_{ref}, X_{cal}) < 0.1$, alors qu'on obtenait $Fcomp_{moy}(X_{ref}, X_{cal}) = 1.97$ en utilisant la méthode de Horaud telle quelle sur 2 couples de données.

D'autre part, une valeur importante des fonctions $Fglob_{moy}$ ou $Fglob_{max}$ n'est pas forcément signe d'une mauvaise détermination de la matrice X . Ainsi pour un bruit de 0.5 *mm*, les valeurs de $Fglob_{moy}$ ou $Fglob_{max}$ sont relativement importantes : approximativement 2.0 *mm* et 6.2 *mm* ; pourtant la matrice X_{cal} trouvée est très proche de la matrice de référence X_{ref} : $Ferr_{moy} \approx 0.3$ *mm* et $Fcomp_{max} \approx 0.5$ *mm*.

Notons également qu'un bruit de l'ordre du millimètre semble une hypothèse réaliste (répétabilité du robot majorée par 1.7 *mm* et précision de l'optotrak de l'ordre de 0.8 *mm*) ce qui nous laisse envisager des résultats assez médiocres avec des erreurs moyennes de l'ordre de 0.75 *mm* et des erreurs maximales de l'ordre du millimètre.

5.4.2 Données réelles

La méthode précédente appliquée à 60 couples de données réelles nous fournit les résultats suivants :

$$\begin{aligned}
 Fglob_{moy}(X_{Shiu}) &= Fglob_{moy}(X_{Horausud}) = 0.37 \text{ mm} \\
 Fglob_{max}(X_{Shiu}) &= Fglob_{max}(X_{Horausud}) = 1.20 \text{ mm}
 \end{aligned}$$

On remarque que les données réelles engendrent des erreurs similaires à celles obtenues sur des données simulées avec un bruit de 0.1 mm ($Fglob_{moy} = 0.37\text{ mm}$ et $Fglob_{max} = 1.20\text{ mm}$). Ce bon résultat est surprenant : on aurait pu penser que les données réelles étaient obtenues avec une précision bien moindre, au vu de la répétabilité et de la précision du robot et de l'optotrak. Rappelons qu'avec un tel bruit (0.1 mm), l'erreur par rapport à la véritable matrice X était en moyenne de 0.05 mm (pour une initialisation des matrices selon la méthode de Horaud) et au pire de 0.08 mm . On peut donc dire que la précision dans le positionnement des marqueurs sur le bras du robot est meilleure que 0.1 mm .

5.5 Remarques et conclusions

Le calibrage du marqueur, c'est-à-dire la détermination de la matrice de passage entre R_{seg} et R_{marq} , peut être effectué en même temps que le calibrage du robot, mais ici on a voulu s'en affranchir pour être indépendant du type de robot utilisé et de la méthode employée pour le calibrer. D'autre part, comme on a plusieurs marqueurs à calibrer, utiliser les résultats du calibrage du robot aurait impliquer deux approches possibles :

- Soit utiliser autant de changeurs de coordonnées du robot qu'il y a de marqueurs. Cette méthode est envisageable car elle n'affecte pas la précision du système, mais elle implique des calculs supplémentaires au cours du suivi, par rapport à la méthode retenue.
- Soit utiliser l'un des changeurs de coordonnées et assigner aux matrices de passage entre R_{seg} et R_{marq} des valeurs fausses, puisque obtenues avec un changeur différent de celui employé par la suite. La solution proposée est meilleure puisque le calcul de la matrice de passage entre R_{seg} et R_{marq} est effectué avec le changeur de coordonnées retenu.

L'étape de calibrage des marqueurs est bien sûr dépendante du modèle du robot. Cependant, cette dépendance n'est pas gênante car la valeur des résidus permet de rendre compte de la qualité du modèle du robot. Si dès cette étape, on détecte un problème avec le robot, il est inutile d'envisager de l'utiliser tel quel pour des tâches exigeant de la précision.

Il faut donc choisir un seuil au delà duquel on considèrera que le robot n'est pas assez précis. Le choix de ce seuil de validité est primordial car la précision du calibrage du marqueur, et donc la précision qu'on peut attendre du système de contrôle redondant complet¹, en dépendent directement.

1. En effet, l'erreur détectée par le système de contrôle redondant provient en partie d'une erreur réelle (celle qu'on veut détecter) et en partie des erreurs accumulées par chacune des phases du système (calibrage du marqueur, des caméras et détection des marqueurs dans les images). Le seuil de détection d'erreur, c'est-à-dire le seuil à partir duquel le système décide qu'il y a une erreur, doit donc être supérieur à la combinaison des imprécisions de chacune des phases du système afin de ne pas détecter des erreurs à tort.

Chapitre 5. Le calibrage des marqueurs

Si le calibrage des marqueurs a été effectué correctement, le système complet de contrôle redondant permettra de détecter au cours de l'évolution du robot, d'une part des incidents techniques, tels qu'un codeur cassé ou une mauvaise transmission des valeurs des codeurs, d'autre part une éventuelle modification des paramètres effectifs du robot par rapport à ceux déterminés au moment du calibrage.

On a vu au paragraphe 5.2.3.4 que les résultats obtenus par les méthodes de Horaud et Shiu étaient peu différents lorsqu'elles étaient appliquées dans de bonnes conditions. Dans notre cas, l'utilisation de l'une ou l'autre est d'autant plus indifférente que la différence des résultats est atténuée par la minimisation qui suit.

Dans notre étude, nous n'avons implémenté que des méthodes qui travaillent sur deux couples de données. Ceci nous a permis de traiter de la même manière les résultats obtenus par l'une ou l'autre des méthodes en cherchant à les optimiser par la minimisation d'une fonction d'erreur globale. Cependant, il serait intéressant de comparer les résultats obtenus par notre minimisation globale, avec les résultats de méthodes qui raisonnent directement sur un ensemble important de données [TL88, ZS93, Wan92].

Enfin, les tests ont montré que la stratégie adoptée fournit une précision meilleure que 0.1 mm dans le positionnement des marqueurs sur le bras d'un robot PUMA 260.

Chapitre 6

Détection et suivi des marqueurs

Grâce aux informations fournies par le robot (valeurs des codeurs) et aux matrices de passages entre les différents repères (connues par calibrage), on va prévoir la position 3D théorique des marqueurs.

Le contrôle redondant consiste alors à vérifier grâce aux images fournies par les caméras que cette position est réellement atteinte (à une certaine marge d'erreur près). La vérification se fait en détectant les marqueurs dans les images (paragraphe 6.2), à partir de la prédiction de leur position (paragraphe 6.1). Grâce au calibrage des caméras, on peut alors associer une droite de rétroprojection à chaque projection d'un marqueur détecté dans une des images. La distance entre cette droite et la position 3D prédite donne une indication de la cohérence des résultats.

Nous avons également étudié succinctement le suivi des marqueurs dans une séquence d'images (paragraphe 6.3).

6.1 Prédiction de la position d'un marqueur

La phase de prédiction de la position d'un marqueur dans une image combine les résultats de toutes les phases préliminaires de calibrage (voir figure 6.1) :

- Le calibrage du marqueur permet de connaître sa position dans R_{seg-i} (le marqueur étant porté par le $i^{ème}$ segment du robot). On peut manipuler le marqueur au moyen d'une matrice de passage (comme lors du calibrage), ou au moyen d'un certain nombre de points caractéristiques. Cette dernière représentation étant plus appropriée à la détection, c'est celle que nous retenons ici¹. Le calibrage du marqueur permet donc de connaître la position de ses points P caractéristiques dans $R_{seg-i} : (P)_{Rseg-i}$.
- Le calibrage intrinsèque du robot permet de connaître la position du marqueur dans R_{robot} en utilisant les valeurs articulaires fournies par le bras : $(P)_{Rrobot} = M_{Rseg-i \rightarrow Rrobot} \cdot (P)_{Rseg-i}$.

1. Le passage d'une représentation à l'autre n'est pas difficile.

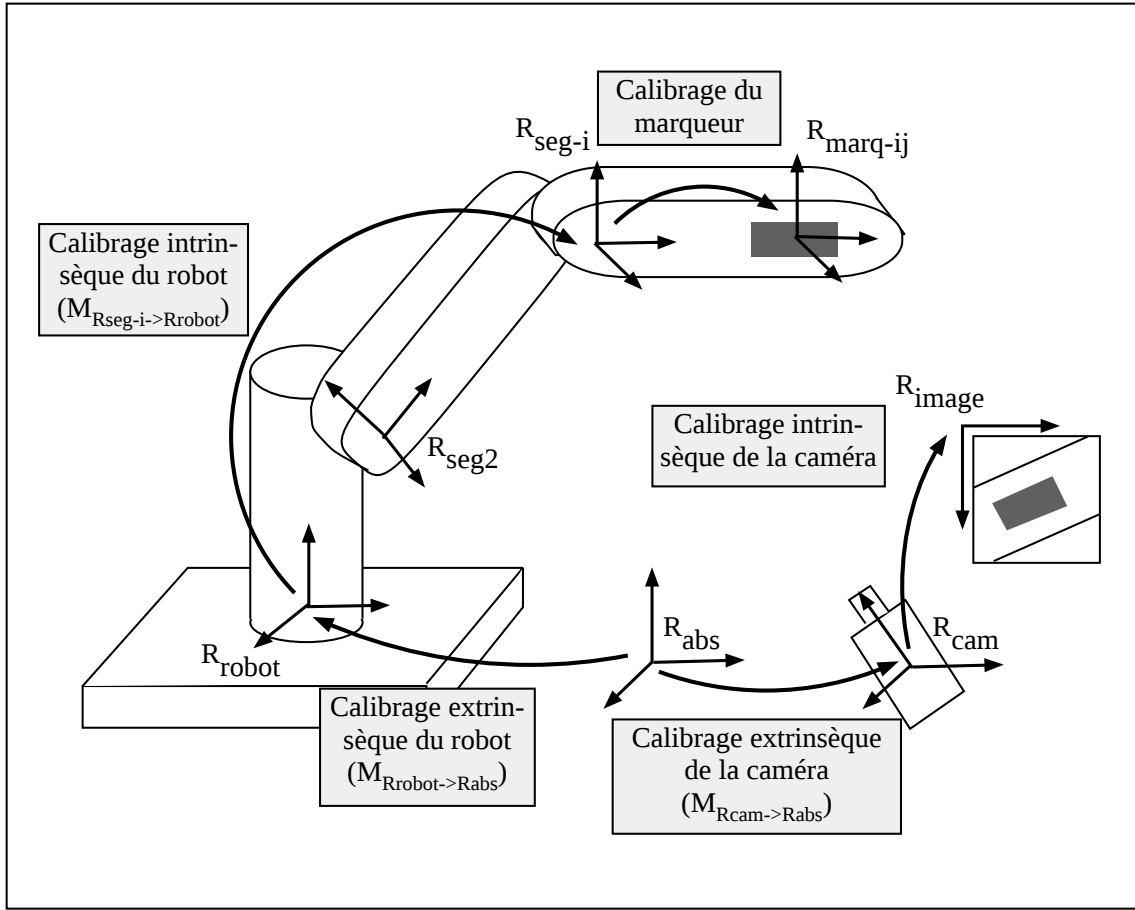


FIG. 6.1 - Prédiction de la position du marqueur

- Le calibrage extrinsèque du robot permet de connaître la position du marqueur dans R_{abs} : $(P)_{R_{abs}} = M_{R_{robot} \rightarrow R_{abs}} \cdot (P)_{R_{robot}}$.
- Le calibrage extrinsèque de la caméra permet de connaître la position du marqueur dans R_{cam} : $(P)_{R_{cam}} = M_{R_{cam} \rightarrow R_{abs}}^{-1} \cdot (P)_{R_{abs}}$.
- Enfin la projection (fonction inverse de la rétroprojection obtenue par calibrage intrinsèque de la caméra) permet de prédire la position du marqueur dans l'image. Notons que la projection n'est pas très précise. Ce n'est pas important puisqu'elle sert uniquement à réduire la fenêtre de recherche du marqueur dans l'image.

La phase de détection pourra alors s'effectuer dans une fenêtre restreinte, ce qui optimisera le temps de calcul. Cette prédiction permet également de minimiser les risques de fausses détections ou d'ambiguïtés.

6.2 Détection d'un marqueur

Nous allons analyser l'influence de la précision de la détection sur la précision du positionnement 3D.

Nous en déduirons qu'il est nécessaire d'obtenir une précision meilleure que le pixel ce qui nous amènera à étudier les algorithmes de détection subpixel et la précision que l'on peut en attendre.

Nous déciderons alors de la forme des marqueurs et de l'algorithme adéquat pour les détecter.

6.2.1 Influence de la précision de la détection sur la précision du positionnement 3D

Remarque préliminaire

Notre objectif n'est pas réellement d'effectuer de la reconstruction. En effet, cela ne serait pas toujours possible à cause de problème de visibilité, et cela nous empêcherait de tirer parti des mesures obtenues lorsqu'un marqueur n'est visible que d'une seule caméra. Aussi nous préférons faire des vérifications de cohérence en évaluant la distance entre la position 3D théorique de chaque marqueur et chacune des droites de rétroprojection qui sont associées à son image², plutôt qu'en évaluant la distance entre la position 3D théorique de chaque marqueur et sa position 3D reconstruite. Cette approche offre plus de souplesse et exploite toutes les informations visuelles dont on dispose.

Cependant implicitement, autoriser une erreur de ϵ_{droite} entre la position 3D prédite et la droite de rétroprojection associée revient à autoriser une erreur de $\epsilon_{reconstruction}$ entre la position 3D prédite et une position 3D reconstruite.

Or si l'on veut éviter des déviations trop importantes, c'est $\epsilon_{reconstruction}$ qu'il faut majorer car c'est elle qui représente la véritable erreur de position 3D. C'est pourquoi nous allons étudier comment une erreur de détection dans l'image se traduit en erreur de reconstruction 3D.

Influence de la précision de la détection sur la précision du positionnement 3D

D'après la figure 6.2, on a :

$$\epsilon_{reconstruction} \approx \frac{\epsilon_{droite}}{\sin\left(\tan\left(\frac{B}{2D}\right)\right)} \approx 2.\epsilon_{droite} \frac{D}{B} \quad \begin{array}{l} \text{pour } D \text{ grand par rapport} \\ \text{à } B, \text{ par exemple } D > 2B \end{array}$$

A titre indicatif, pour $D = 2 \text{ m}$ et $B = 1 \text{ m}$, on obtient $\epsilon_{reconstruction} \approx 4.\epsilon_{droite}$.

2. ou à ses images, s'il est visible de plusieurs caméras

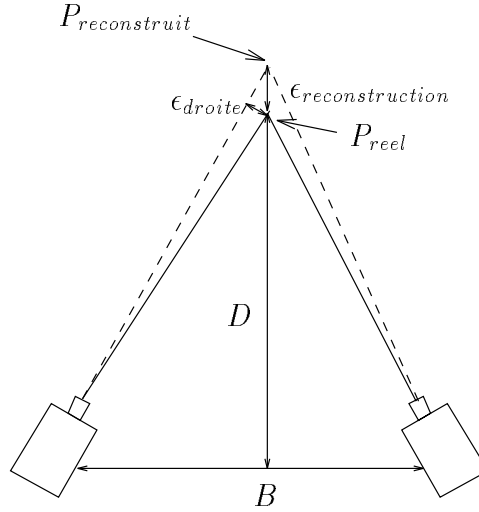


FIG. 6.2 - Erreur de reconstruction 3D engendrée par des erreurs de détection

D'autre part, pour un champ observé d'environ 1 m dans une image discrétisée selon 512 pixels, chaque pixel représente environ 2 mm .

Avec les mêmes valeurs que précédemment, une erreur de 1 pixel lors de la détection du marqueur dans l'image, se traduit donc par une erreur $\epsilon_{droite} \approx 2\text{ mm}$ et par une erreur $\epsilon_{reconstruction} \approx 8\text{ mm}$, à laquelle il faut ajouter l'erreur issue du modèle de calibrage des caméras.

Ainsi il est intéressant d'obtenir une détection avec une précision meilleure que le pixel pour limiter les erreurs en 3D.

6.2.2 Etude bibliographique des algorithmes de détection subpixel

Les approches abordées ici sont de nature différente de celles de la segmentation. La segmentation consiste à déterminer avec la résolution du pixel, des limites d'objets, ou bien des zones qui correspondent à des objets, sans avoir d'information riche sur la nature de ces objets, même si des connaissances a priori sont très souvent exploitées.

Dans notre cas, on sait parfaitement ce qu'on cherche : dans une zone restreinte de l'image, on veut déterminer précisément, c'est-à-dire avec une résolution meilleure que le pixel, la position d'un motif particulier que l'on définira ultérieurement, constitué d'éléments géométriques présentant un fort contraste avec le fond.

L'utilisation combinée de ces contraintes portant sur un bon nombre de pixels, permet d'affiner la précision de la localisation.

La plupart des méthodes de détection subpixel répertoriées exploitent la richesse des informations a priori en construisant un modèle de ce que l'on cherche, puis dé-

terminent les paramètres de ce modèle (dont les paramètres de position/orientation qui seuls nous intéressent) pour le mettre en correspondance avec les données réelles de l'image. Ces données peuvent être pré-traitées et donc se présenter sous des formes plus ou moins complexes.

Nous allons essayer de recenser les représentations possibles de l'image à traiter, les modèles et les méthodes de détermination et d'optimisation des paramètres présents dans la littérature.

6.2.2.1 Les représentations d'images

image binaire

La représentation la plus élémentaire est une image binaire construite à partir de l'image de départ : les pixels sans intérêt valent 0 ; les pixels estimés appartenir à l'objet cherché (qui peut être un objet à proprement parler, ou bien un contour) valent 1.

Cette image binaire peut être obtenue par des méthodes de segmentation classiques (détection de contrastes, croissance/décroissance de régions ...).

image brute

L'image est exploitée telle quelle.

image continue

La manière la "plus exacte" d'obtenir une image continue repose sur l'hypothèse que les niveaux de gris de l'image brute sont une discrétisation d'un signal continu correspondant à l'image. Les techniques de traitement du signal permettent d'obtenir la formule de reconstruction (voir [Pic80] pour plus de détails). On explicite ci-dessous la reconstruction en 1D pour plus de simplicité, mais le principe est bien sûr le même en 2D :

Soit $x(t)$ un signal continu nul en dehors de $[0, T]$. On peut le développer en série de Fourier :

$$x(t) = \sum_n x_n \cdot \exp(-in\omega t) \quad \text{avec} \quad \omega = \frac{2\pi}{T}$$

$$\text{et} \quad x_n = \frac{1}{T} \int_0^T x(t) \exp(in\omega t) dt = \frac{1}{T} \int_0^T x(t) \exp\left(\frac{i2\pi tn}{T}\right) dt$$

Soit $\tilde{x}(n)$ la transformée de Fourier de $x(t)$. Supposons que $\tilde{x}(n)$ soit nulle en dehors de $[-B, B]$.

Chapitre 6. Détection et suivi des marqueurs

Alors on peut également développer $\tilde{x}$ en série de Fourier :

$$\tilde{x}(\nu) = \sum_n \tilde{x}_n \cdot \exp\left(\frac{-i2\pi\nu n}{2B}\right) \quad \text{et} \quad \tilde{x}_n = \frac{1}{2B} \int_{-B}^B \tilde{x}(\nu) \cdot \exp\left(\frac{i2\pi\nu n}{2B}\right) d\nu$$

Or l'intégrale est la transformée de Fourier inverse de $\tilde{x}$ pour $t = n/2B$. Il s'agit donc de $x(n/2B)$.

$$\text{d'où} \quad \tilde{x}_n = \frac{1}{2B} \cdot x\left(\frac{n}{2B}\right).$$

On peut donc reconstituer $\tilde{x}(n)$ à partir des valeurs discrètes $x(t_n)$ où $t_n = n/2B$. Puis par transformée de Fourier inverse :

$$\begin{aligned} x(t) &= \int_{-B}^B \tilde{x}(\nu) \cdot \exp(i2\pi\nu t) d\nu \\ &= \int_{-B}^B \left[\sum_n \left(\frac{1}{2B} \cdot x(t_n) \right) \exp(-i2\pi\nu t_n) \right] \exp(i2\pi\nu t) d\nu \end{aligned}$$

d'où la formule d'échantillonnage :

$$x(t) = \sum_n x(t_n) \frac{\sin(2\pi B(t - t_n))}{2\pi B(t - t_n)} = \sum_n x(t_n) \operatorname{sinc}(2\pi B(t - t_n))$$

Cette formule n'est valable que si la fréquence du signal est inférieure à la fréquence d'échantillonnage ($f_{\text{echan}} > f_{\text{Nyquist}} = B$).

Cependant, la fonction sinc a un support infini. Il faudrait donc utiliser tous les points $x(t_n)$ pour calculer $x(t)$, ce qui est impossible. Aussi, on utilise généralement une fonction d'interpolation à support fini, telle que le plus proche voisin (qui n'est pas adapté ici, puisqu'on veut améliorer la précision !), une interpolation linéaire ou une interpolation par B-spline cubique. Comme ces fonctions ont des réponses en fréquence différentes de la fonction sinc, la restauration du signal par l'une de ces fonctions peut s'en trouver altérée.

[TH86] a comparé ces trois fonctions dans le domaine des fréquences et a montré que la B-spline cubique a la meilleure réponse. Cependant, des expérimentations avec une interpolation linéaire ont montré qu'elle engendrait une erreur (0.005 pixel) bien inférieure à celle engendrée par le bruit du système (0.05 pixel). Ce type d'interpolation semblerait donc suffisant dans la plupart des cas. En fait, cette remarque est valable pour l'utilisation qui en est faite dans [TH86] (corrélation de deux images rééchantillonnées), mais elle ne s'applique pas à des traitements de type "recherche d'extremum".

image rééchantillonnée

Il s'agit simplement de la rediscrétisation d'une image continue selon des intervalles plus petits que ceux de l'image brute.

Une fois ces nouvelles images construites, on peut leur appliquer les méthodes classiques de segmentation. [TH86] applique ainsi une fonction de cross-corrélation entre deux images rééchantillonnées.

En théorie, la précision de la détection s'en trouve améliorée proportionnellement à l'augmentation de la résolution de l'image par rapport à l'image initiale. Néanmoins, la précision ainsi atteinte dépend de la précision avec laquelle la nouvelle image approxime l'image originale (voir [PKD83] pour plus de détails).

On peut procéder dans un ordre différent, c'est-à-dire traiter d'abord l'image initiale en lui appliquant un filtre, puis rééchantillonner l'image filtrée.

C'est ce qui est fait par [HM86]. Ils convoluent l'image initiale avec un masque LoG ("Laplacian of Gaussian"). Puis ils construisent une image continue grâce à un modèle polynomial du second ordre appliqué à des régions 3 x 3. Cette image continue est ensuite rediscrétisée selon une résolution 2 ou 3 fois meilleure que celle de l'image initiale. Enfin ils déterminent les zéros de cette nouvelle image, qui correspondent aux points de contraste de l'image initiale. Notons qu'ils auraient pu chercher les zéros dans l'image continue, ce qui aurait sans doute amélioré la précision, mais ralenti le processus.

6.2.2.2 Les modèles

La complexité du modèle retenu est liée à la complexité de la représentation d'images. Ainsi, il est bien sûr inutile de construire un modèle très sophistiqué si l'on ne dispose que d'une image binaire ...

modèle purement géométrique

C'est le type de modèles le plus simple. Il n'exploite que l'aspect purement géométrique des motifs cherchés. Pour une droite, il aura 2 paramètres (la pente θ et la distance à l'origine ρ) : $\sin \theta . u - \cos \theta . v = \rho$. Pour un cercle, il aura 3 paramètres, les coordonnées du centre (u_c, v_c) et le rayon R : $(u - u_c)^2 + (v - v_c)^2 = R^2$...

modèle construit à partir de ce que sont censés représenter les niveaux de gris des pixels de l'image

Le niveau de gris d'un pixel représente la quantité de lumière reçue par la partie du capteur qui correspond à ce pixel.

On peut donc construire un modèle des niveaux de gris d'une portion d'image dans laquelle se trouve un objet dont on connaît les paramètres de forme et de position. Le niveau de gris d'un pixel donné est une combinaison linéaire des deux niveaux de gris correspondant au fond de la scène et à l'objet, dans la proportion des surfaces occupées. Il dépend donc des paramètres de forme et de position de l'objet.

Ainsi, soient g_{objet} , le niveau de gris de l'objet, g_{fond} , le niveau de gris du fond ; soient p_k , les paramètres qui définissent l'objet ; alors pour tout pixel (i, j) , son

niveau de gris théorique est donné par :

$$g(i, j) = g_{\text{objet}} \cdot \alpha(i, j, p_k) + g_{\text{fond}} \cdot (1 - \alpha(i, j, p_k))$$

où $\alpha(i, j, p_k)$ est la proportion de la surface du pixel (i, j) occupé par l'objet.

Selon la nature de l'objet, α peut être calculée de manière analytique, ou bien de manière approximative, en discrétisant le pixel en un certain nombre de points répartis régulièrement, puis en dénombrant les points situés à l'intérieur de l'objet.

Cependant cette représentation n'est pas exacte. On peut en voir un contre-exemple sur la figure 6.3 dans le cas d'une ligne de contraste verticale.

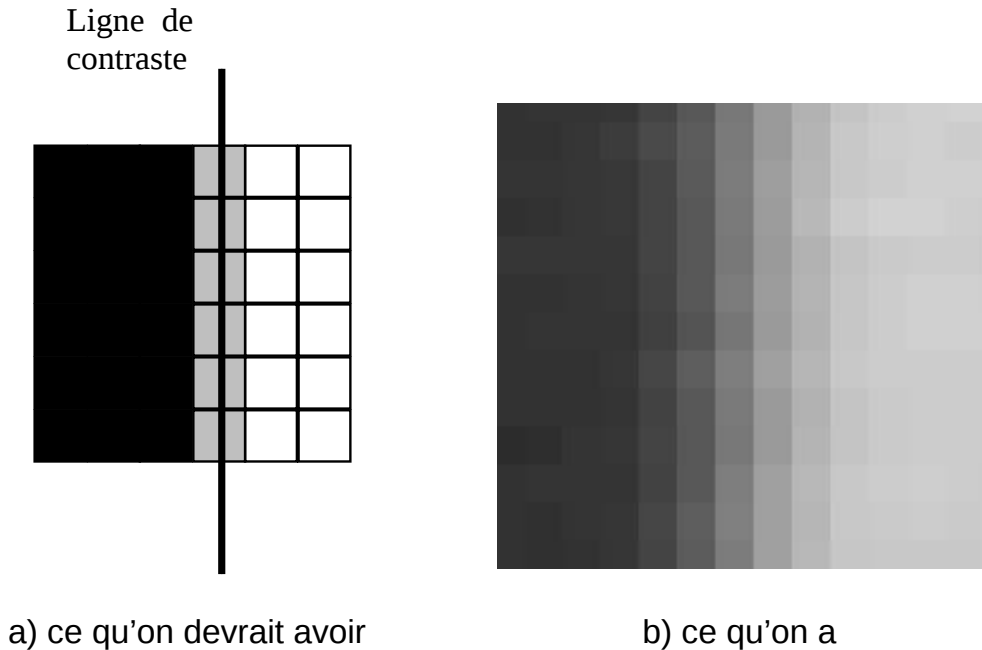


FIG. 6.3 - L'image digitalisée d'une ligne de contraste verticale b) ne correspond pas au modèle théorique a)

On constate un étalement des niveaux de gris intermédiaires sur plusieurs colonnes (au lieu d'une seule colonne dans le cas idéal du modèle). Ceci peut s'expliquer par un mauvais réglage de la mise au point de la caméra qui engendre une image floue, mais aussi par le processus d'acquisition d'image. En effet, chaque ligne du capteur CCD est d'abord convertie en un signal analogique par la caméra (voir figure 6.4) pour être transmise, via un câble vidéo, à la station de travail. Avant de pouvoir être exploitée, ce signal est rééchantillonné par un numériseur qui recrée une représentation digitale de la ligne (voir figure 6.5).

Ces différentes transformations ont tendance à agir comme un filtre passe-bas, supprimant les hautes fréquences et donc atténuant la pente du contraste.

Ces phénomènes n'étant pas pris en compte par le modèle précédemment décrit, certains auteurs ont cherché à construire des modèles plus complexes pour les intégrer. Ces modèles sont basés sur une hypothèse gaussienne du flou.

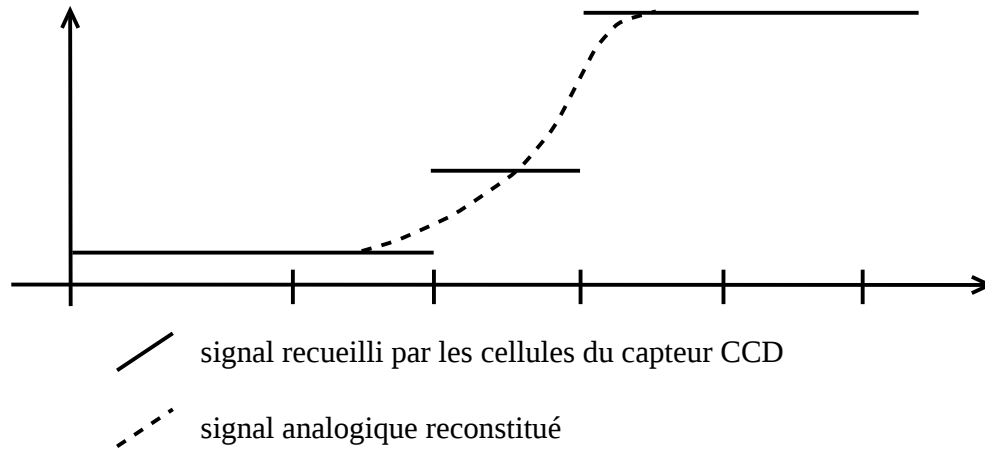


FIG. 6.4 - *Création d'un signal analogique à partir des données d'une ligne de capteur CCD*

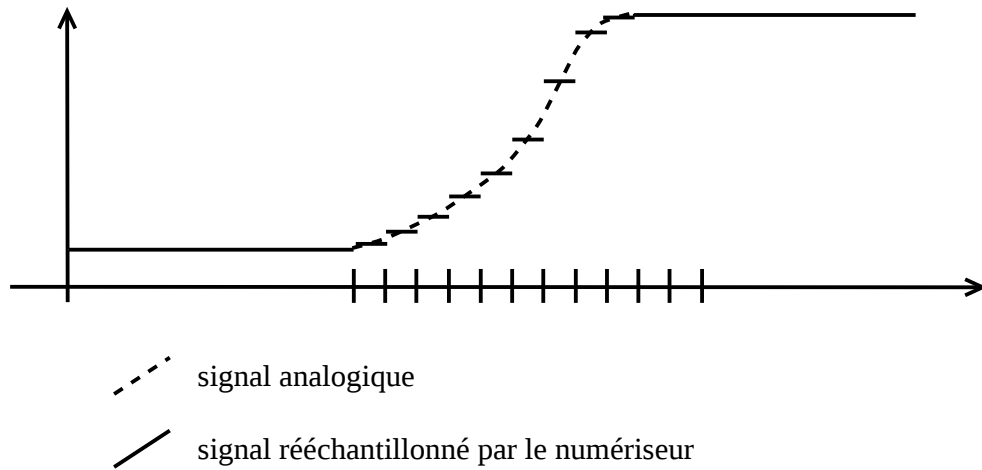


FIG. 6.5 - *Numérisation du signal analogique*

modèles intégrant un flou gaussien

[Peu93] a étudié différentes fonctions mathématiques susceptibles d'approximer le signal en niveaux de gris dans le voisinage d'un indice visuel. De manière empirique, ce sont les fonctions exponentielles qui sont apparues les plus proches de la réponse des systèmes d'acquisition.

Ces fonctions sont d'autant plus intéressantes qu'elles présentent un plan asymptotique qui peut traduire une variation d'intensité lumineuse dans la scène.

Les fonctions proposées par [Peu93] sont de la forme :

$$\begin{aligned} F_1(x, y) &= a + bx + cy + d \exp(-g^2 y^2) && \text{pour une droite} \\ F_2(x, y) &= a + bx + cy + d \exp(-f^2 x^2 - g^2 y^2) && \text{pour un point} \\ F_3(x, y) &= a + bx + cy + d \exp(-f^2 x^2) + e \exp(-g^2 y^2) && \text{pour une croix} \\ &&& (2 \text{ droites perpendiculaires}) \end{aligned}$$

Il faut rajouter aux paramètres a, b, c, d, e, f et g , une translation (u_0, v_0) et une rotation θ pour obtenir une expression de ces fonctions dans le repère de l'image :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & -u_0 \\ \sin(\theta) & \cos(\theta) & -v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

On pourrait également rajouter un autre paramètre pour modéliser des croix dont les deux branches ne seraient pas perpendiculaires ... Tous ces modèles portent sur des indices visuels "fins" qui apparaissent sur un fond.

D'autres modèles [BM94] portent sur la transition entre deux surfaces homogènes. On distingue deux régions homogènes R_1 et R_2 de niveau de gris A et B et une région intermédiaire R_3 constituée d'un flou gaussien des 2 autres régions.

Soit $g(\alpha)$, la fonction gaussienne :

$$g(\alpha) = \frac{1}{\sqrt{\pi\sigma^2}} \exp\left(-\frac{\alpha^2}{\sigma^2}\right)$$

Considérons à titre d'exemple, le cas d'un modèle de coin. Soient (x_0, y_0) ses coordonnées exactes. Alors on a (voir [BM94]) :

$$Im(x, y) = \begin{cases} A & \text{si } (x, y) \in R_1 \\ B & \text{si } (x, y) \in R_2 \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(\alpha) g(\beta) I'(x - \alpha, y - \beta) d\alpha d\beta & \text{si } (x, y) \in R_3 \end{cases}$$

$$\text{avec } I'(x, y) = \begin{cases} A & \text{si } x \geq x_0 \text{ et } y \geq y_0 \\ B & \text{sinon} \end{cases}$$

Aux paramètres A et B , il faut ajouter de la même manière que précédemment les paramètres de rotation, translation et éventuellement un paramètre indiquant l'angle pour un coin non carré.

modèles statistiques

Un modèle statistique permettrait de tenir compte des effets du bruit dans la création d'images. [HD83] décrit un tel modèle : il suppose une certaine distribution statistique pour les niveaux de gris de chaque côté de la ligne de contraste (par exemple des distributions gaussiennes définies par leurs moyennes g_{droite} et g_{gauche} et leurs variances σ_{droite} et σ_{gauche}) ; alors chaque pixel contenant une partie de la droite de contraste a un niveau de gris qui est un mélange des niveaux de gris droit et gauche, c'est-à-dire qu'il a une moyenne et une variance qui dépendent de la proportion de la surface du pixel à gauche de la droite et de g_{droite} , g_{gauche} , σ_{droite} et σ_{gauche} .

image précédente

Les modèles précédents ont recours à plus de paramètres que ceux qu'on veut déterminer (position et orientation uniquement) car on manque d'information sur l'image.

En revanche, lors de l'étude d'une séquence d'images, on dispose des informations obtenues grâce aux images précédentes (en particulier, les niveaux de gris des différentes régions).

Ces informations peuvent être utilisées de manière classique, comme initialisation des paramètres à chercher lors d'un processus d'optimisation.

On peut également utiliser l'image précédente de manière brute, et chercher uniquement les paramètres de translation et de rotation entre une région de l'image précédente et une région de l'image courante [MRRO94]. Ceci n'est valable que si le déplacement d'une image à l'autre est suffisamment faible, pour ne pas engendrer de grosses transformations.

Cette manière de procéder permet de déterminer un déplacement relatif par rapport à l'image précédente et non pas une position absolue.

6.2.2.3 Les méthodes de détermination et d'optimisation des paramètres

Ces méthodes dépendent bien sûr du modèle et du type d'image dont on dispose. Elles ne sont donc pas applicables dans tous les cas.

La méthode générale consiste à ajuster les paramètres du modèle de telle sorte que le modèle soit le plus proche possible des données réelles. On cherche un nombre très restreint de paramètres (θ et ρ pour une droite, ou une ligne de contraste ; (x_c, y_c) et éventuellement R pour un cercle ou un disque ...) alors que l'on dispose d'un grand nombre d'équations potentielles (égal au nombre de pixels sur lesquels

Chapitre 6. Détection et suivi des marqueurs

on veut mettre en correspondance les données réelles et le modèle). On doit donc résoudre un système surcontraint.

On peut procéder de plusieurs manières :

- procéder à un vote sur l'ensemble des paramètres possibles ;
- combiner ces informations pour réduire le nombre d'équations et raisonner directement sur des informations globales ;
- optimiser les paramètres sur l'ensemble des données en minimisant une fonction "différence" ou en maximisant une fonction "ressemblance".

Vote

Ce type de méthodes est appliqué à des images binaires et à des modèles géométriques. L'avis de chacun des pixels retenus (de valeur "1" dans l'image binaire) est pris en compte pour éliminer ou pour plébisciter, selon la méthode, un ensemble de n-uplets de paramètres.

Élimination des paramètres incohérents avec l'une des données

[BKLO87] traitent le cas de la détermination d'une droite de cette manière. Ils disposent d'une droite numérisée D_{num} , c'est-à-dire d'un ensemble de pixels par lesquels passe la droite "réelle" cherchée, $D_{cherchee}$. Chaque pixel de D_{num} impose une contrainte supplémentaire sur la droite cherchée (à savoir que $D_{cherchee}$ doit passer par le pixel courant), ce qui réduit l'espace des droites possibles et donc améliore la précision de $D_{cherchee}$.

De manière similaire, à partir d'un ensemble de pixels appartenant au contour d'un cercle, [Hav91] détermine une région possible (dénommée "locale") pour son centre. [EG94] ont repris cette méthode et montrent qu'une majoration du diamètre de cette région est de l'ordre de $R^{-\frac{1}{2}}$ où R est le rayon du cercle dont on cherche le centre. Donc, si l'on choisit un point de cette région (le barycentre par exemple), comme centre du cercle, une majoration de la précision atteignable sera de l'ordre de $R^{-\frac{1}{2}}$.

Cependant, ce procédé est trop restrictif puisqu'il ne tolère pas d'erreur lors de la binarisation de l'image. Il suffit d'une donnée aberrante pour éliminer les bons paramètres.

Plébiscite des paramètres cohérents avec la donnée courante

Plutôt qu'un vote éliminatoire, il vaut mieux utiliser un vote majoritaire, dans lequel chaque pixel retenu vote pour l'ensemble des n-uplets de paramètres cohérents avec lui. C'est le principe des méthodes de type "transformée de Hough" [Hou62].

Pour des raisons d'implémentation, l'espace des paramètres est discrétisé. A la fin du vote, on retient le n-uplet discret ayant remporté le plus de suffrages.

Les méthodes reposant sur la transformée de Hough sont d'autant plus précises que l'image a été rééchantillonnée avec une grande résolution avant d'être prétraitée pour en faire une image binaire, et surtout que la discrétisation de l'espace des paramètres est fine. Cependant, le fait d'affiner la discrétisation de l'espace des paramètres augmente le temps d'exécution de l'algorithme. En revanche, il est possible de restreindre davantage l'ensemble des paramètres cohérents en ajoutant des contraintes supplémentaires portant par exemple sur l'orientation du gradient [Bal81] (on a alors besoin de données plus riches qu'une simple image binaire).

L'un des inconvénients pratiques des méthodes basées sur la transformée de Hough, est qu'il y a souvent plusieurs n -uplets contigus qui obtiennent un bon score. Il peut être intéressant de les combiner pour moduler les résultats. Ceci devient même indispensable si l'on cherche plusieurs instances du même type d'objet dans une même image, les k n -uplets ayant les scores maximaux ne représentant pas forcément k instances différentes.

Enfin, notons que ces algorithmes ont été étendus à des formes non nécessairement paramétrées [Bal81, Dav82].

Création d'informations globales

Pour localiser le contraste d'un bord de type marche en 1D (voir figure 6.6), [TM84] utilise les moments statistiques évalués sur une fenêtre de longueur n pixels. Le moment d'ordre i est défini par :

$$m_i = \frac{1}{n} \int_0^n g(x)^i dx \quad \text{où } g(x) \text{ est le niveau de gris pour l'abscisse réelle } x$$

Ayant 3 paramètres à déterminer (h , k et l), il calcule les moments d'ordre 1, 2 et 3 sur son modèle de marche pour obtenir les formules théoriques des moments en fonction des paramètres du modèle, ainsi que sur son image pour obtenir leurs valeurs courantes. Il peut ainsi établir 3 équations qui lui permettent d'identifier les 3 paramètres avec en particulier, l , la localisation de la marche.

[LMAR89] effectue la même démarche avec les moments spatiaux :

$$m_i = \frac{1}{n} \int_0^n g(x).x^i dx$$

Contrairement aux moments statistiques, les moments spatiaux ne présentent pas de biais en présence de bruit (à condition malgré tout que la fenêtre de calcul soit centrée sur le pixel auquel appartient le bord de la marche, ce que l'on peut obtenir en calculant une première fois les paramètres, puis en recentrant la fenêtre de calcul pour une deuxième estimation).

Le calcul des moments permet également d'estimer la position d'un signal de type "impulsion". Il suffit au préalable d'intégrer le signal pour obtenir un signal de type "marche" (figure 6.7).

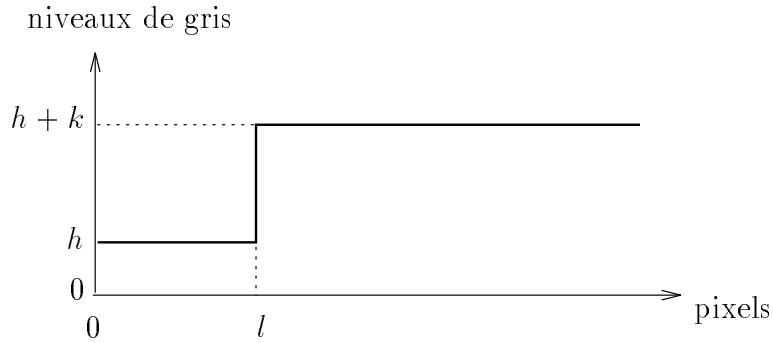


FIG. 6.6 - *Modèle de type "marche"*

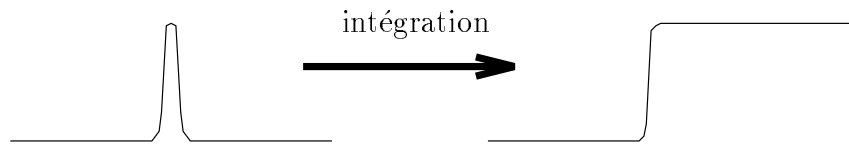


FIG. 6.7 - *Intégration d'un signal de type "impulsion"*

Ces méthodes sont extensibles à des images 2D.

Une manière simple de procéder consiste à appliquer la méthode 1D soit aux lignes, soit aux colonnes de l'image pour obtenir un ensemble de points dont on peut déduire une droite (une estimation a priori de l'orientation de la marche permettra de choisir dans quelle direction appliquer l'opérateur 1D).

On peut également calculer les moments 2D dans une fenêtre circulaire. En raisonnant sur la transformation de ces moments lorsqu'on applique une rotation à la fenêtre, [LMAR89] donne une méthode directe de calcul des paramètres h , k , l et θ de la marche.

Pour déterminer précisément le centre d'une région fermée ou d'une surface, le calcul du centroïde (ou barycentre) est fréquemment utilisé. Ce calcul peut porter uniquement sur les points de contour ou bien sur l'ensemble des pixels de la surface. Chaque terme peut également être pondéré (par exemple par le niveau de gris du pixel dans le cas de surface).

Il s'agit en fait d'une version simplifiée de la méthode des moments spatiaux 2D, dans le sens où elle ne nécessite que le calcul des 2 moments d'ordre 1, et fournit directement la position du centre. Cette simplification provient du fait que la forme traitée est symétrique³. Cette méthode ne s'applique donc qu'à ce type de formes.

3. On définit une forme symétrique de la manière suivante : une forme est symétrique s'il existe un point, le centre, tel qu'il soit le milieu de tout segment le contenant et reliant 2 points du

Dans le cadre d'images binaires, [BI90] ont étudié différentes formes pour obtenir une précision de mise en correspondance subpixel. Le choix d'une forme compacte (donc un disque par excellence) réduit les erreurs dues au bruit sur la périphérie des marqueurs. Les diamants et carrés se révèlent être de mauvais candidats, à cause de leurs cotés rectilignes qui biaisent la détection. Les disques sont en revanche bien adaptés grâce à leur périmètre courbe qui fournit une distribution plus uniforme des pixels partiellement inclus à l'intérieur.

La détection du centre d'un disque est d'autant plus précise que son rayon est important : dans [EG94], il est montré que le calcul du centroïde permet de déterminer le centre d'un disque binaire avec une précision en $\mathcal{O}(R^{-\frac{1}{2}})$ en moyenne. Dans le cadre d'images non binaires, il faut traduire les niveaux de gris de telle sorte que le niveau de gris du fond soit nul, afin de ne pas biaiser le résultat en décalant le centroïde vers le centre de la fenêtre de calcul (on peut également procéder comme en 1D, en effectuant plusieurs estimations après avoir recentré la fenêtre de calcul).

Toutes ces méthodes sont simples à mettre en oeuvre et permettent d'obtenir des résultats directement. Cependant, elles n'utilisent que des données globales (les moments) et sont donc forcément moins précises que des méthodes qui optimisent les paramètres sur l'ensemble des données.

Optimisation de paramètres

Cette optimisation de paramètres peut être abordée de 2 manières différentes. Il s'agit :

- soit de maximiser une fonction de ressemblance entre le modèle et l'image,
- soit de minimiser une fonction d'erreur entre le modèle et l'image.

Maximisation d'une fonction de ressemblance

[Dvo83] calcule la fonction de corrélation discrète entre l'image à traiter et une imagelette de référence. Il interpole les valeurs de cette fonction par une surface et cherche à en déterminer le maximum. Il utilise pour cela un estimateur quadratique dont la précision dépend de la manière dont la fonction de corrélation autour du pic peut être approximée par une parabole.

Si l'on a construit un modèle statistique, on peut estimer la probabilité qu'un pixel ait un niveau de gris donné. On cherche alors les paramètres (il s'agit d'une droite dans le cas de [HD83]) qui maximisent la vraisemblance d'observer les niveaux de gris $g_1, g_2 \dots g_n$ présents dans l'image réelle

Minimisation d'une fonction d'erreur

périmètre de la forme. Par définition, le centre est unique.

Chapitre 6. Détection et suivi des marqueurs

L'erreur que l'on cherche à minimiser peut être la différence d'une fonction appliquée au modèle et de cette même fonction appliquée aux données réelles (de manière similaire aux moments, qui eux permettaient d'obtenir une solution directe).

Un exemple d'une telle fonction est donné par [KVV94], qui définissent une énergie locale capable d'identifier différents types de contraste, en combinant des filtres symétrique et antisymétrique. L'optimisation des paramètres est ensuite effectuée par un algorithme de descente de gradient.

Si l'on dispose d'un modèle purement géométrique, il semble naturel de minimiser une fonction d'erreur construite à partir de la distance géométrique entre des points de l'image et le modèle. Cette fonction porte sur un ensemble de points pré-calculés par d'autres méthodes. Les coordonnées de ces points peuvent être entières (ce qui revient à travailler avec une image binaire), ou réelles (elles ont alors été évaluées par d'autres méthodes subpixel, par exemple par une détection de contraste de type marche en 1D, ou par la détection d'un extremum en intensité selon les lignes ou les colonnes de l'image, ou selon une direction plus appropriée [BMB94] ...).

Pour effectuer une minimisation aux moindres carrés, la fonction d'erreur est définie de la manière suivante :

$$f(p) = \sum_{i=1}^N d^2(P_i, \mathcal{O}(p))$$

$$\text{où } \begin{cases} N \text{ est le nombre de points considérés} \\ P_i \text{ est le } i^{\text{ème}} \text{ point} \\ p \text{ est l'ensemble des paramètres courants (à optimiser)} \\ \mathcal{O}(p) \text{ est l'objet instancié avec les paramètres courants} \\ d \text{ est la distance euclidienne} \end{cases}$$

Pour minimiser cette fonction, on doit résoudre un système linéaire traduisant que la dérivée de f par rapport à chacun des paramètres est nulle.

Dans le cas de la recherche d'une droite, la minimisation de la fonction précédente revient à déterminer la droite de régression linéaire.

Dans le cas d'un cercle, on aboutit au résultat par l'intermédiaire de la pseudo-inverse du système linéaire engendré. Cette solution n'est pas satisfaisante car elle implique l'inversion d'une matrice de taille N .

[Lan87] propose une formulation vectorielle du problème qui aboutit à la résolution itérative de 2 équations. Il fournit une évaluation de la vitesse de convergence de son algorithme en fonction de l'amplitude de l'arc dont il dispose (il n'est pas indispensable que les points décrivent un cercle complet). L'inconvénient de cette méthode est qu'elle ne fournit pas une solution directement.

Aussi [TC89] reconsidèrent le problème grâce à l'utilisation d'une "distance" différente. A la distance euclidienne, ils préfèrent une distance surfacique, ce qui semble intéressant puisqu'on travaille en 2D. Cette approche permet d'obtenir une solution directe, mais présente l'inconvénient d'être biaisée. Une formulation du biais est fournie, ce qui ne permet malheureusement pas de le corriger, puisqu'il dépend

bien sûr de la variance du bruit dans les données. Cependant, le biais ne semble pas très important pour des cercles complets (0.3 unités pour un cercle complet de rayon 100 unités avec un bruit de 5 unités).

[Jos94] montre les inconvénients des méthodes décrites dans [Lan87] et [TC89] en présence d'un bruit important et propose un algorithme itératif plus stable.

Si l'on dispose d'un modèle en niveaux de gris, une fonction d'erreur "naturelle" sera :

$$f(p) = \sum_{i=1}^N (g_{image}(i) - g_{modele}(i, p))^2$$

où $\left\{ \begin{array}{l} N \text{ est le nombre de pixels de la région considérée} \\ p \text{ est l'ensemble des paramètres courants (à optimiser)} \\ g_{image}(i) \text{ est le niveau de gris du } i^{ème} \text{ pixel de l'image} \\ g_{modele}(i) \text{ est le niveau de gris du } i^{ème} \text{ pixel du modèle instancié avec les} \\ \text{paramètres courants} \end{array} \right.$

Pour minimiser cette fonction, Pascal Brand emploie l'algorithme de Levenberg-Marquard [PFTV92] aussi bien pour des modèles de croix identiques à ceux de Peuchot [Peu93] que pour des modèles de coins [BM94].

Hyde et Davis [HD83] font remarquer qu'un algorithme de descente standard ne fonctionnerait pas bien avec leur fonction, car ses dérivées partielles présentent des discontinuités dues aux coins des pixels (cette remarque ne s'applique pas aux modèles de Brand qui sont eux $\mathcal{C}^\infty$). Ils utilisent donc un algorithme de type "steepest descent" qui n'a pas recours aux dérivées de la fonction.

Toutes ces méthodes peuvent bien sûr être combinées :

- Tichem et Cohen [TC94] font une première estimation du centre d'un disque par le centroïde, puis détectent précisément des points du contour le long de demi-droite partant du centroïde. Les 30 points du contour ainsi déterminés sont regroupés par 3 (séparés de 120 degrés). Chaque groupe permet de calculer un centre, le résultat final étant la moyenne des 10 centres ainsi trouvés.
- Brand, Mohr et Bobet [BMB94] cherche la position d'une croix comme l'intersection de 2 droites. Pour déterminer précisément un point d'une droite, ils cherchent le point le plus sombre le long d'une droite à peu près perpendiculaire à la droite cherchée. Cette droite perpendiculaire est discrétisée selon l'algorithme de Bresenham. Les niveaux de gris de cette discrétisation sont interpolés par une B-spline cubique dont on cherche le minimum pour localiser le point le plus sombre de la droite. Cette opération est effectuée pour une trentaine de points différents. La droite cherchée est celle qui passe au mieux, au sens des moindres carrés, par ces points.
- ...

6.2.3 Précision

6.2.3.1 Perte de précision due à la numérisation

L'un des problèmes dus à la numérisation des images provient, comme on l'a déjà vu, des signaux ayant des fréquences trop élevées par rapport à la fréquence d'échantillonnage du système d'acquisition.

Une solution pour résoudre ce problème consiste à lisser préalablement les deux images à mettre en correspondance (i.e l'image à traiter et le modèle) pour supprimer les effets engendrés par les hautes fréquences.

Un autre problème est dû au nombre fini de bits utilisés pour représenter un pixel (8 bits en général). Pour évaluer l'ampleur de ce phénomène, [TH86] a synthétisé une image contenant 2 droites dont il a imposé la position relative. Une première estimation de la distance entre ces 2 droites a été faite sur une image dont les niveaux de gris étaient codés par des réels. il a obtenu une erreur de 0.000005 pixels. L'expérience a été réitérée avec une image dont les niveaux de gris étaient codés sur 8 bits. L'erreur était alors de 0.01 pixels.

Enfin demeurent les bruits dus au système d'acquisition d'images. [TH86] acquiert deux images successivement et mesure une erreur de 0.03 pixel entre les deux.

Nous avons effectué le même genre de mesures (voir au paragraphe 3.2.2) et nous avons également estimé des écarts de l'ordre de 0.03 pixel en moyenne.

Il est vain de vouloir obtenir des précisions meilleures que la répétabilité des systèmes d'acquisition. Mais comment évaluer la précision atteinte sur des images réelles ?

6.2.3.2 Evaluation de la précision

Plusieurs auteurs ont fait des mesures sur des images synthétisées [BI90, TC89, Jos94, Lan87], mais on peut difficilement modéliser le bruit introduit par les systèmes d'acquisition. D'autres auteurs ont déduit, de manière théorique, des majorations et des ordres de grandeur de la précision atteignable, en fonction du bruit présent dans les données, des caractéristiques du motif qu'ils cherchait à localiser et de l'algorithme utilisé [LMAR89, TM84, EG94], mais il est difficile d'en tirer des renseignements pour une image réelle si l'on ne connaît pas la quantité de bruit présent.

Il serait donc intéressant de pouvoir mesurer expérimentalement une telle précision. Très peu d'études ont été faites dans ce sens. La principale difficulté de ce genre de mise en oeuvre provient de la méconnaissance des caractéristiques exactes des systèmes d'acquisition d'images. La précision de la détection dans une image est donc intimement liée à la précision du calibrage du système d'acquisition, d'autant plus que la qualité de ce calibrage repose elle-même en grande partie sur la précision de la détection d'indices visuels.

[TC94] comparent les centres d'un disque et d'un tore concentriques (à la précision de fabrication près qui correspond dans leurs expérimentations à 0.1 pixel). Les décalages mesurés sont inférieurs à 0.15 pixel après avoir ajusté itérativement l'inclinaison du motif de test par rapport au système d'acquisition d'images.

Peuchot [Peu93] utilise deux grilles entrelacées, dont il détecte précisément les croisements. La première grille permet de calibrer la caméra. La seconde sert à mesurer les erreurs combinées de la détection des croix et du calibrage. Peuchot annonce une variance de 0.01 pixel dans la localisation et une erreur maximale inférieure à 0.03 pixel pour une caméra CCD standard.

Egalement avec une grille, [BMB94] comparent les corrections qu'ils apportent à l'image pour tenir compte des distorsions. D'une image à l'autre, la différence quadratique moyenne des corrections est de 0.03 pixel et la différence maximale de correction est de 0.08 pixel pour la méthode utilisant des B-splines pour détecter les croisements.

Quatre méthodes d'évaluation de la précision de la détection d'un coin sont décrites dans [BM94].

La première mesure l'alignement d'un certain nombre de coins.

Une deuxième méthode consiste à établir une relation entre la variance du bi-rapport (que l'on peut mesurer expérimentalement) et la variance des données, en l'occurrence les coins détectés.

Ces deux méthodes risquent de surestimer l'erreur de détection car elles travaillent sur l'image brute et n'essaient pas de corriger les éventuelles distorsions induites par le système d'acquisition d'images.

Le calcul des bi-rapports permet néanmoins de vérifier que la détection n'est pas biaisée.

Les deux autres méthodes reposent sur la comparaison de résultats obtenus à partir de données réelles et à partir de données simulées dont on connaît le bruit. Les résultats en question sont, d'une part des erreurs de reconstruction 3D, d'autre part des distances entre les points détectés et les droites épipolaires associées.

Ces quatre méthodes fournissent des résultats très similaires, à savoir une variance d'environ 0.1 pixel.

6.2.4 Conclusions

6.2.4.1 Choix de la forme du marqueur

Pour faire des évaluations d'erreurs rapides, il est plus simple de manipuler des points (ou non pas des formes complexes). Ces points seront estimés à partir de formes plus "tangibles" afin d'être extraits des images. Ces formes doivent être suffisamment simples pour qu'on puisse utiliser des algorithmes rapides, afin d'envisager un suivi temps réel. Le marqueur sera donc constitué de motifs géométriques élémentaires (cercles, droites ...).

Chapitre 6. Détection et suivi des marqueurs

Ainsi nous avons besoin de localiser précisément des points. On pense donc à des marqueurs constitués de disques. L'algorithme du centroïde est simple à implémenter, rapide et peu sensible aux erreurs dues à la numérisation, surtout si le disque est suffisamment grand.

Cependant, la projection d'un disque à travers une caméra, est une ellipse. Et le centre de l'ellipse ne correspond pas à la projection du centre du cercle. La différence entre ces points (centre de l'ellipse et projection du centre du cercle) est d'autant plus grande que le rayon du cercle est grand et que l'angle formé par l'axe optique et la normale au plan contenant le disque est important (voir l'annexe B pour une estimation de la différence entre le centre de l'image d'un segment et la projection du centre de ce même segment).

Dans le cas où l'inclinaison du disque par rapport à l'axe optique de la caméra est important, on est confronté à une contradiction : si l'on veut une bonne détermination du centroïde, le rayon du disque doit être grand, et si l'on veut que l'effet de perspective n'engendre pas une grosse erreur dans la localisation du centre, le rayon du disque doit être petit.

Les marqueurs à base de disques sont donc rejetés.

Néanmoins, les disques sont des motifs très intéressants dans le cadre d'une mire de calibrage placée à peu près perpendiculairement à l'axe optique de la caméra (ce qui est notre cas).

Un autre moyen simple de définir un point est de le considérer comme l'intersection de deux droites.

On décide d'utiliser 3 croix afin de définir un marqueur : les 3 points désignés par chacune des croix permettent d'associer un repère au marqueur.

Notons que les croix sont identifiées du fait qu'on prévoit l'endroit où chacune d'entre elle va se projeter dans l'image. Il n'y a donc pas d'ambiguïté lors de la construction du repère.

Dans l'image, les croix doivent être suffisamment grandes pour être détectées. Des tests nous ont montré que des branches de longueur 10 pixels et de largeur 2 pixels conviennent bien à la détection. Dans la suite, nous considérerons cette condition comme un minimum à respecter.

En ce qui concerne la taille des croix réelles, la condition précédente se traduit en fonction des conditions d'utilisation (zoom, distance d'observation). Elle devra donc être déterminée pour chaque application. Néanmoins, elle devra également respecter des contraintes d'encombrement.

Les choix pratiques concernant les marqueurs (matériau, taille, nombre, disposition) sont discutés au paragraphes 8.1.3.

6.2.4.2 Algorithme retenu

L'algorithme doit être suffisamment rapide pour permettre d'envisager un suivi temps réel. [HD83] a comparé un algorithme linéaire (en travaillant sur une image binaire) et un algorithme non linéaire (intégrant les niveaux de gris) : les résultats

s'avèrent peu différents alors que l'algorithme non linéaire est bien plus coûteux en temps de calcul.

Il ne semble donc pas indispensable de “traquer le centipixel” d'autant plus que le marqueur que l'on doit détecter bouge durant le temps d'exécution de la détection. Le gain de précision obtenu par un algorithme coûteux serait annihilé par la perte de précision due à l'aspect dynamique du suivi.

Nous avons retenu l'algorithme décrit dans [BMB94] à base de B-splines. L'exécution de cet algorithme nécessite de connaître approximativement la position du centre de la croix, ainsi que les orientations des deux droites. Le centre de la croix est bien sûr l'intersection des deux droites qui la composent. Chaque droite est détectée de la manière suivante :

- Autour du centre approximatif de la croix, on parcourt une imagerie selon une direction perpendiculaire à l'orientation de la droite cherchée.
- Pour chaque segment de pixels ainsi parcouru, on construit une fonction spline qui approxime les niveaux de gris des pixels rencontrés.
- On cherche le maximum de cette fonction et on retient ce point si son niveau de gris paraît acceptable (par exemple s'il est supérieur à 75% du niveau de gris le plus élevé de l'imagerie). Pour l'étape de maximisation, nous avons utilisé l'algorithme de la section dorée décrit dans [PFTV92].
- On calcule la droite de régression linéaire sur l'ensemble des points retenus.

Cet algorithme est peu coûteux : la détermination des paramètres des deux droites qui composent la croix est faite par une simple régression linéaire sur un ensemble de points. Il est cependant très précis, les points étant eux-mêmes détectés par la recherche du maximum d'une fonction d'interpolation spline, le long d'une droite perpendiculaire à la droite cherchée. La construction de la spline et la recherche du maximum sont ici rapides puisqu'on travaille en une seule dimension.

Nous n'avons pas fait d'évaluation de la précision de cet algorithme. Une telle évaluation ne peut d'ailleurs pas être indépendante du calibrage intrinsèque de la caméra, à moins d'être effectuée sur des images synthétisées. Or pour synthétiser une telle image, il est nécessaire de recourir à un modèle de l'image d'une croix qui ne correspondra pas exactement à la réalité. Néanmoins une telle évaluation serait très intéressante.

Nous étudierons des résultats plus globaux, intégrant la détection des croix et les calibrages intrinsèque et extrinsèque de deux caméras, au chapitre 7.

6.3 Suivi d'un marqueur dans une séquence d'images

Nous allons d'abord considérer le cas où le suivi peut s'effectuer dans des conditions idéales, c'est-à-dire lorsque le marqueur concerné est parfaitement visible des

caméras qui l'observent. Nous envisagerons alors le cas où le marqueur est progressivement occulté. Nous serons amenés à modifier l'apparence des marqueurs afin d'anticiper plus facilement l'apparition d'une telle occultation. Enfin nous aborderons le cas où il n'est pas possible d'exploiter d'autres informations a priori que celles extraites des images précédentes.

6.3.1 Suivi effectué dans des conditions idéales

Dans des conditions idéales, la surveillance d'un marqueur peut se faire en bouclant sur l'ensemble des actions suivantes :

1. prédiction de la position 3D des points caractéristiques du marqueur $P_{3D_{prédit}}$,
2. prédiction de la position des points caractéristiques du marqueur dans l'image $P_{2D_{prédit}}$, par projection de leur position 3D prédite $P_{3D_{prédit}}$,
3. détection et localisation précise des points caractéristiques du marqueur dans l'image $P_{2D_{détecté}}$,
4. calcul des droites de rétroprojection $\mathcal{D}$ associées aux points caractéristiques du marqueur,
5. vérification de la cohérence par calcul de la distance entre les points caractéristiques 3D prédits $P_{3D_{prédit}}$ et les droites de rétroprojection $\mathcal{D}$ associées à leurs images respectives.

6.3.2 Prise en compte des occultations

La manière de procéder décrite au paragraphe précédent présente un inconvénient majeur : si la phase de détection ne s'effectue pas correctement, il est impossible de décider si cela provient d'une simple occultation du marqueur ou d'une véritable erreur.

On peut remédier à ce problème en exploitant la continuité de l'acquisition des images pour prédire l'apparition d'une occultation. Il est en effet peu probable qu'un marqueur disparaisse complètement entre deux images successives. La disparition se faisant progressivement, une prédiction d'une occultation totale est tout-à-fait envisageable.

Pour faciliter cette prédiction, nous proposons de modifier l'apparence des marqueurs de manière à pouvoir quantifier l'occultation. La figure 6.8 présente le marqueur que nous avons retenu pour l'instant :

- Le contour extérieur est constitué de segments de droites présentant un fort contraste avec le fond. Ces contours peuvent être détectés de la même manière que les branches des croix des anciens marqueurs. Les points caractéristiques du marqueur sont les quatre coins du contour extérieur.

- L'intérieur du marqueur est un damier constitué de $4n^2$ cases dont la position est connue par rapport aux quatre points caractéristiques.

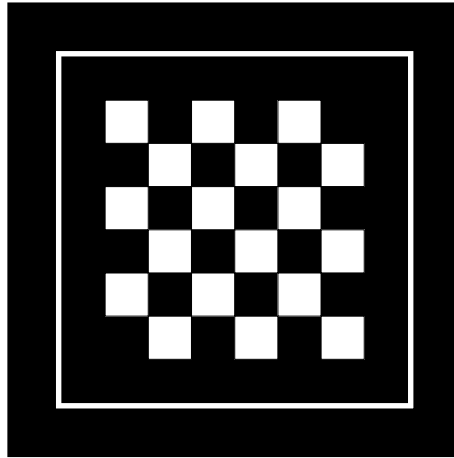


FIG. 6.8 - Marqueur facilitant la prévision d'une occultation

Si l'on parvient à détecter un nombre suffisant de droites du contour extérieur et de points caractéristiques (au moins trois morceaux de droites et deux points caractéristiques), on peut compter de manière très simple le nombre de carrés visibles : on prévoit grossièrement l'emplacement du centre de chaque carré et l'on vérifie que le niveau de gris du pixel associé correspond à ce que l'on attend (niveau faible pour les carrés noirs, élevé pour les carrés blancs). Si la vérification est bonne, on considère ce carré comme valide. Afin de ne pas biaiser le décompte si le marqueur est caché par une surface blanche ou noire, on ne retient que les carrés valides et dont tous les voisins immédiats⁴ sont valides.

Le nombre de carrés retenus fournit un “degré d'occultation du marqueur”, c'est-à-dire une indication sur la proportion de la surface du marqueur qui est occultée. L'évolution du degré d'occultation d'un marqueur au cours du suivi permet d'anticiper la disparition complète du marqueur. Ainsi en cas d'absence du marqueur dans l'image, on peut en distinguer la cause : simple occultation ou véritable erreur. Notons qu'une erreur brutale peut être confirmée par l'examen des valeurs renvoyées par les codeurs du robot. D'autre part, dans le cas de caméras mobiles, l'anticipation de la disparition du marqueur permet de reconsidérer la stratégie de surveillance et d'assigner à la caméra concernée la surveillance d'un autre marqueur.

Ainsi, pour être en mesure d'anticiper la disparition du marqueur, sa surveillance se fait en bouclant sur l'ensemble des actions suivantes :

1. prédiction de la position 3D des points caractéristiques du marqueur,
2. prédiction de la position des points caractéristiques du marqueur dans l'image,

4. On appelle “voisins immédiats” d'un carré, les carrés qui ont un côté commun avec lui.

Chapitre 6. Détection et suivi des marqueurs

3. si la détection et la localisation précise des points caractéristiques du marqueur dans l'image ont pu être effectuées alors :

- (a) détermination du degré d'occultation,
- (b) calcul des droites de rétroprojection,
- (c) vérification de la cohérence par calcul de la distance entre les points caractéristiques 3D prédits et les droites de rétroprojection associées,
- (d) éventuellement changement de stratégie (observation d'un autre marqueur) après examen de l'évolution du degré d'occultation.

sinon

- (a) examen de l'évolution du degré d'occultation pour décider si l'absence du marqueur est due à une occultation ou à une erreur,
- (b) s'il s'agit d'une erreur, envoi d'un message d'alarme.

6.3.3 Suivi sans connaissance a priori

Dans les deux cas précédents, on a supposé que les informations renvoyées par les codeurs étaient suffisamment bonnes pour que la prédiction de la position des marqueurs dans les images permettent de les détecter facilement.

Cependant, on peut s'autoriser une marge d'erreur importante lors de la surveillance, pour des phases où la précision de la position du robot n'est pas prépondérante. Dans ce cas, la prédiction de la position des marqueurs dans les images, via la chaîne des résultats de calibrage préliminaires, peut s'avérer insuffisamment précise pour permettre une bonne détection des marqueurs, sans que pour autant la marge d'erreur 3D autorisée soit dépassée. D'autre part, dans le cas où l'on a détecté une panne, il peut également être intéressant de suivre les marqueurs dans les images afin de maintenir un bon contrôle du robot en dépit de la panne (voir paragraphe 8.3.3).

Dans ces deux cas de figure, on risque de perdre les marqueurs dans les images alors que les étapes précédentes pourraient nous fournir des indications sur leur dérive. Ainsi, il semble judicieux d'exploiter l'information acquise grâce à l'image précédente.

Soit $u_{calib}(i)$, la position du marqueur prédite uniquement par les étapes de calibrage, dans la $i^{ème}$ image de la séquence. Soit $u_{prédit}(i)$, la position du marqueur prédite selon diverses méthodes que nous allons examiner. Soit $u_{détecté}(i)$, la position du marqueur détectée dans la $i^{ème}$ image.

La manière la plus simple d'exploiter l'information acquise précédemment, consiste à initialiser la recherche du marqueur dans l'image courante avec sa position dans l'image précédente. On a alors :

$$u_{prédit}(i) = u_{détecté}(i - 1)$$

Cette approche fonctionne uniquement si le déplacement du robot n'est pas trop rapide : le déplacement induit des marqueurs dans les images doit être inférieur à l'erreur de prédiction autorisée par l'algorithme de détection. Dans notre cas, l'algorithme de détection autorise une erreur de prédiction d'environ 3 ou 4 pixels. Pour un champ observé d'environ $50 \times 50 \text{ cm}^2$ avec une digitalisation de 512×512 pixels et une fréquence de traitement de 5 Hz , la vitesse du robot doit être inférieure à 2 cm/s !

Pour améliorer l'initialisation de la recherche du marqueur, on peut prédire son déplacement, plutôt que sa position absolue, et l'ajouter à la position précédemment détectée. Soit $d_{\text{prédit}}(i, j)$ le déplacement prédit entre les images i et j . Il peut être obtenu très simplement comme la différence entre la position prédite par calibrage pour l'image précédente et la position prédite pour l'image courante :

$$d_{\text{prédit}}(i - 1, i) = u_{\text{calib}}(i) - u_{\text{calib}}(i - 1)$$

On a alors :

$$\begin{aligned} u_{\text{prédit}}(i) &= u_{\text{détecté}}(i - 1) + d_{\text{prédit}}(i - 1, i) \\ &= u_{\text{détecté}}(i - 1) + u_{\text{calib}}(i) - u_{\text{calib}}(i - 1) \end{aligned}$$

Dans le cas d'une approche par asservissement visuel comme nous la verrons au paragraphe 8.3.3.3, le déplacement peut être prédit à l'aide de la matrice d'interaction qui décrit les variations des signaux 2D (en l'occurrence la position des marqueurs dans les images) en fonction des variations des valeurs articulaires du robot :

$$d_{\text{prédit}}(i - 1, i) = \tilde{J}(q(i)).\Delta q_i = \tilde{J}(q(i)).(q(i) - q(i - 1))$$

où $\begin{cases} q(j) \text{ est le vecteur des valeurs articulaires du robot au moment de la saisie de l'image } j, \\ \tilde{J} \text{ est la matrice d'interaction restreinte aux lignes qui concernent le marqueur courant.} \end{cases}$

Cette approche utilise encore de la connaissance a priori, à savoir la matrice d'interaction et les valeurs articulaires du robot (ou du moins une estimation de ces valeurs). Dans un cadre différent de notre projet, où l'on ne disposerait que des images précédentes, on pourrait envisager une prédiction de la position courante des marqueurs par des méthodes du type filtrage de Kalman[DF90].

6.3.4 Conclusion

Pour le moment, nous n'avons implémenté que l'algorithme le plus simple décrit au paragraphe 6.3.1. Le projet final nécessitera au moins la prise en compte des occultations. Les expérimentations menées jusqu'à présent ont été réalisées dans des conditions idéales de visibilité. Nous allons les détailler dans le chapitre suivant.

Chapitre 7

Expérimentations

Nous avons effectué deux expérimentations pour évaluer le système dans son intégralité.

La première a été réalisée dans les locaux de l'EDF et a porté uniquement sur la partie vision du système. Les dimensions du dispositif expérimental étaient de l'ordre des dimensions impliquées par une application de type nucléaire.

La seconde expérimentation intègre toutes les étapes nécessaires à une application de type médical.

7.1 Evaluation de la partie VISION dans le cadre de l'application nucléaire

Cette expérimentation a pour objectif d'évaluer la précision relative atteignable par la partie VISION du projet. Nous effectuons de la reconstruction 3D à partir des images fournies par deux caméras.

7.1.1 Dispositif expérimental et matériel utilisé

Les caméras dont nous disposons sont deux modules de caméscope SONY. Ces caméscopes sont à focale et focus variables, mais nous les utilisons de manière fixe, le calibrage à focale variable n'étant pas encore opérationnel au moment des expérimentations. Les expérimentations ont été effectuées avec ces caméras dans le but d'évaluer la précision atteignable avec le matériel de l'EDF déjà prévu pour d'autres tâches dans le bol de générateur de vapeur.

Le marqueur, de taille $145 \times 85 \text{ mm}^2$, est constitué de trois croix blanches sur fond noir. Les croix sont faites d'un revêtement rétro réfléchissant et leurs branches ont une longueur de 18 mm et une épaisseur de 3 mm . Le fond noir est un velours qui absorbe la lumière.

L'EDF a fait confectionner un support sur lequel on peut fixer le marqueur et un "rigid body" optotrak de manière solidaire. On dispose également d'un mannequin

Chapitre 7. Expérimentations

sur lequel on peut amarrer le support du marqueur et du rigid body optotrak en trois positions connues les unes par rapport aux autres (voir photo 7.1).

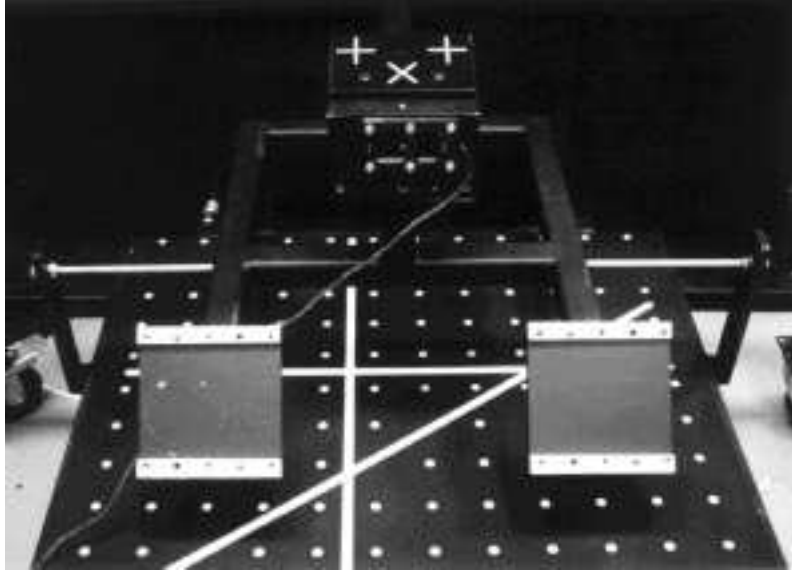


FIG. 7.1 - *Le support du marqueur et du rigid body placé sur l'une des trois positions du mannequin*

Deux caméscopes calibrés observent la scène et localisent le marqueur, c'est-à-dire les trois croix qui le constituent, en chacune des trois positions. Pour chaque croix, on calcule les trois translations que l'on compare, d'une part aux valeurs de référence connues par construction, d'autre part aux valeurs fournies par l'optotrak. La photo 7.2 présente le dispositif expérimental.

7.1.2 Etapes impliquées

Les différentes phases impliquées par cette expérimentation sont décrites sur le schéma 7.3.

- Le **calibrage intrinsèque** de chacun des deux caméscopes consiste à déterminer les transformations, notées abusivement comme des matrices de passage $T_{cam-int_1}$ et $T_{cam-int_2}$, qui à tout pixel (u, v) de l'image i , associent une droite de rétroprojection $\mathcal{D}_i$ dans le repère R_{camera_i} . Nous fixons la valeur focale et le focus des deux caméscopes.

Ce calibrage est effectué par la méthode des deux plans avec une modélisation cubique. La position de chaque plan est repérée uniquement par les valeurs de translation renvoyées par le rail sur lequel est montée la mire de calibrage et l'on fait l'hypothèse que la mire est translatée perpendiculairement à son plan. Les deux plans de calibrage sont distants de 80 cm.

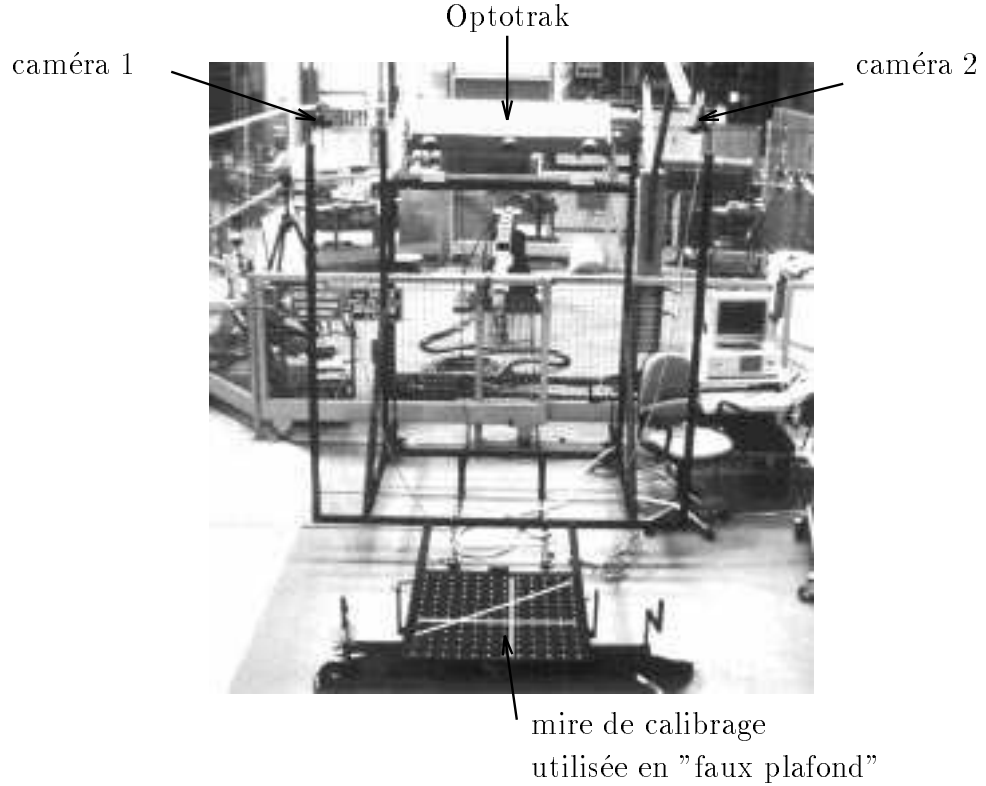


FIG. 7.2 - *Dispositif expérimental*

- Le **calibrage extrinsèque** des caméscopes par rapport au repère absolu consiste à déterminer les matrices de passage $T_{cam-ext_1}$ et $T_{cam-ext_2}$. Pour simuler le plafond du bol de générateur de vapeur, nous avons utilisé la mire de calibrage. La détection des “embouchures des tubes” dans les images s’en trouve simplifiée, sans que la méthode de recalage ne soit modifiée. Il s’agit de déterminer pour chaque caméscope, la transformation qui minimise la somme des distances au carré entre chaque trou et la droite de rétroprojection associée à son image (voir paragraphes 4.1.1 et 4.1.3).
- La **mesure de la position 3D du marqueur** consiste à déterminer T_{obs} , la position du marqueur dans le repère $R_{faux-plafond}$. En fait nous estimons, dans les images, la position 2D de chacune des trois croix qui constituent le marqueur, selon la procédure décrite au paragraphe 6.2.4.2. La position 3D que nous retenons est l’intersection, au sens des moindres carrés, des droites de rétroprojection calculées à partir des images 1 et 2 : chaque droite est obtenue par détection de la croix dans l’image i , puis calcul de la droite de rétroprojection dans le repère R_{camera_i} grâce au calibrage intrinsèque, et enfin calcul de cette même droite dans le repère $R_{faux-plafond}$ grâce au calibrage extrinsèque.

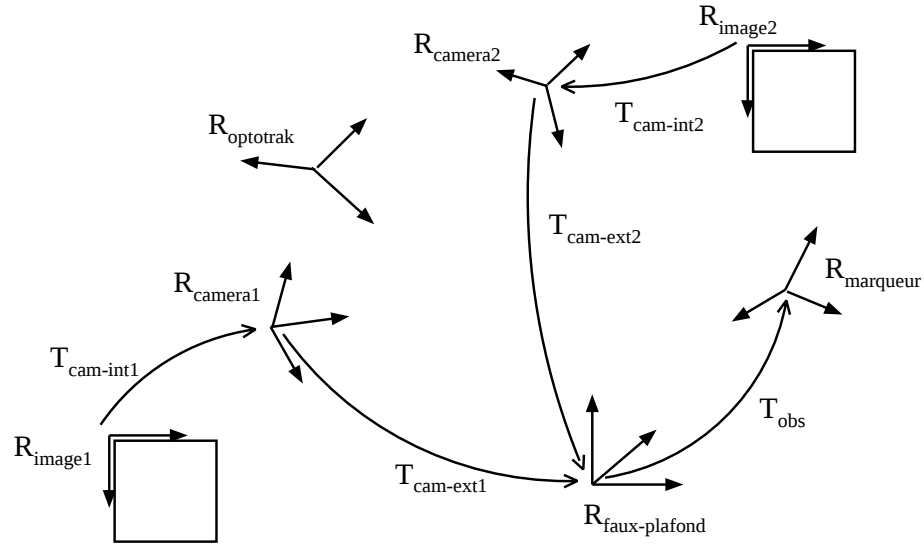


FIG. 7.3 - Les différents repères impliqués

7.1.3 Résultats

	par construction	par optotrak	par la vision	
distance en <i>mm</i> entre les positions 1 et 2	346.41	346.88	croix 1	349.71
			croix 2	349.88
			croix 3	349.54
distance en <i>mm</i> entre les positions 1 et 3	346.41	346.40	croix 1	349.02
			croix 2	349.05
			croix 3	349.08
distance en <i>mm</i> entre les positions 2 et 3	400.00	399.99	croix 1	400.86
			croix 2	400.74
			croix 3	401.13

Les résultats obtenus par l'optotrak sont excellents. Ils sont bien meilleurs que ne le laissait présager la précision annoncée au paragraphe 5.2.2.2.

Les erreurs commises par la vision sont de l'ordre de 1%. Le protocole expérimental a cependant présenté deux lacunes qui peuvent expliquer cette précision modérée: d'une part, les supports des caméras étaient "flexibles" et peuvent donc avoir subi de légères oscillations; d'autre part, l'un des caméscopes s'est avéré peu répétable ce qui a impliqué des paramètres de longueur focale et de focus légèrement différents entre la phase de calibrage intrinsèque et les phases de calibrage extrinsèque et de mesure des positions des croix. Par manque de disponibilité du matériel de l'EDF, nous n'avons malheureusement pas pu procéder à une nouvelle expérimentation en conservant les mêmes réglages pour les calibrages intrinsèque et

extrinsèque des caméras et pour la phase d'acquisition d'images.

Ces résultats ont cependant satisfait les responsables de l'EDF qui envisagent de surveiller un robot Shilling dont la précision est au mieux de 3 mm en moyenne à vide, et de 5 mm chargé.

7.2 Evaluation du système complet dans le cadre de l'application médicale

Cette expérimentation a pour but d'intégrer toutes les étapes du système de contrôle redondant et d'évaluer la précision relative atteignable par la partie VISION qui comprend le calibrage des caméras et la détection des marqueurs dans les images, et par la partie PREDICTION ROBOTIQUE qui comprend le calibrage du robot et le calibrage des marqueurs.

7.2.1 Dispositif expérimental et matériel utilisé

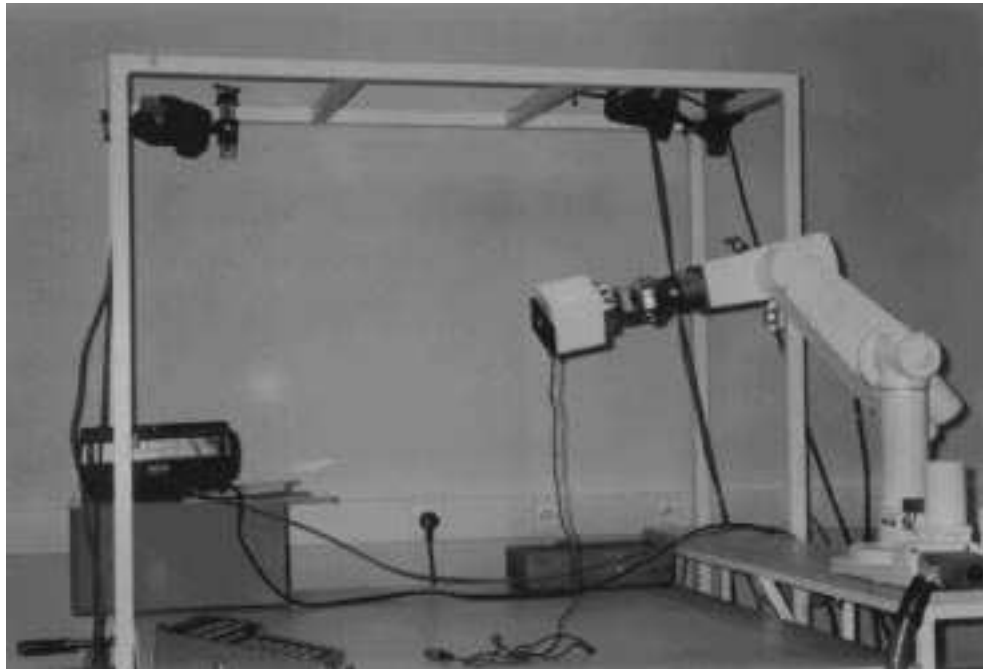


FIG. 7.4 - *Dispositif expérimental*

On dispose de :

- Un robot : PUMA 260
- Un cube cible : ce cube de 8 cm de côté est porté par le robot (photo 7.5).

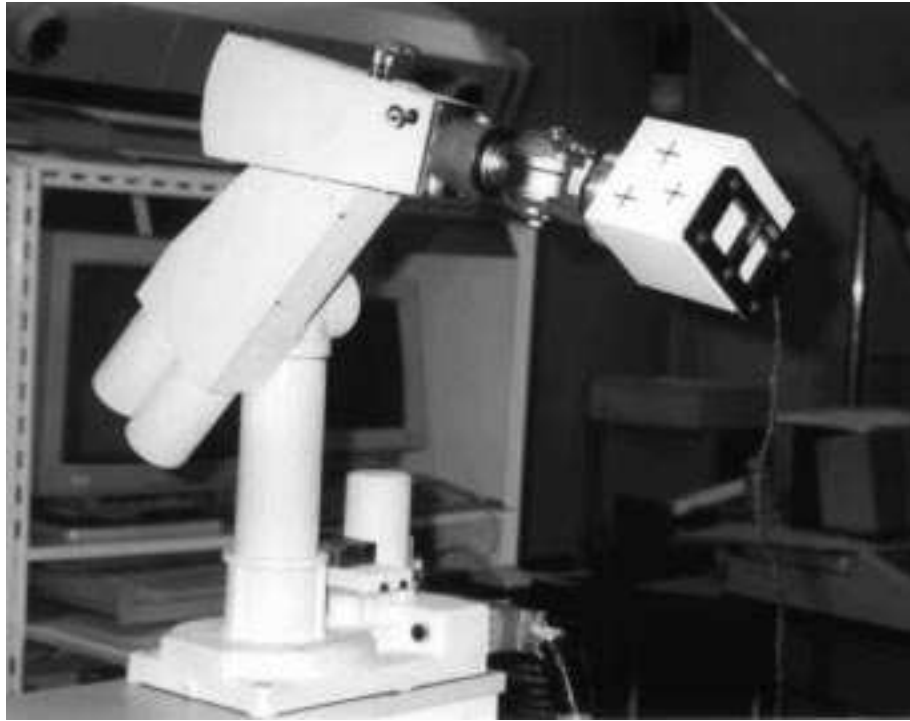


FIG. 7.5 - *Le cube porté par le robot*

Un marqueur, constitué de 3 croix noires sur fond blanc, est placé sur l'une de ses faces. Les croix sont de simples décalcomanies et leurs branches ont une longueur de 8.5 mm et une épaisseur de 1.5 mm .

Un “rigid body” optotrak est fixé sur une autre des faces du cube. Ce “rigid body” présente une double utilité : d'une part, il sert au calibrage du marqueur par rapport au robot ; d'autre part, il fournit une mesure optotrak que l'on peut comparer avec les résultats de la méthode développée ici.

- Deux caméras JVC TK-S300 : elles sont situées à environ 2.5 m de la cible ; elles ont un objectif de 25 mm , ce qui leur permet d'englober un champ commun d'environ $L \times H \times P = 20\text{ cm} \times 20\text{ cm} \times 30\text{ cm}$.
- Les 3 caméras optotrak. Les mesures fournies par l'optotrak nous servent ici de références. L'optotrak est également utilisé lors des phases préliminaires de calibrage des caméras et de calibrage du marqueur.

7.2.2 Etapes impliquées

Le schéma 7.6 présente les divers repères et transformations impliqués dans le cadre d'un système complet.

Nous avons choisi comme repère absolu, le repère associé au robot. D'autre part, le calibrage des caméras se fait selon la méthode décrite au paragraphe 4.2.2 grâce à une mire de calibrage portée par le robot. Les repères R_{camera_1} et R_{camera_2} sont donc

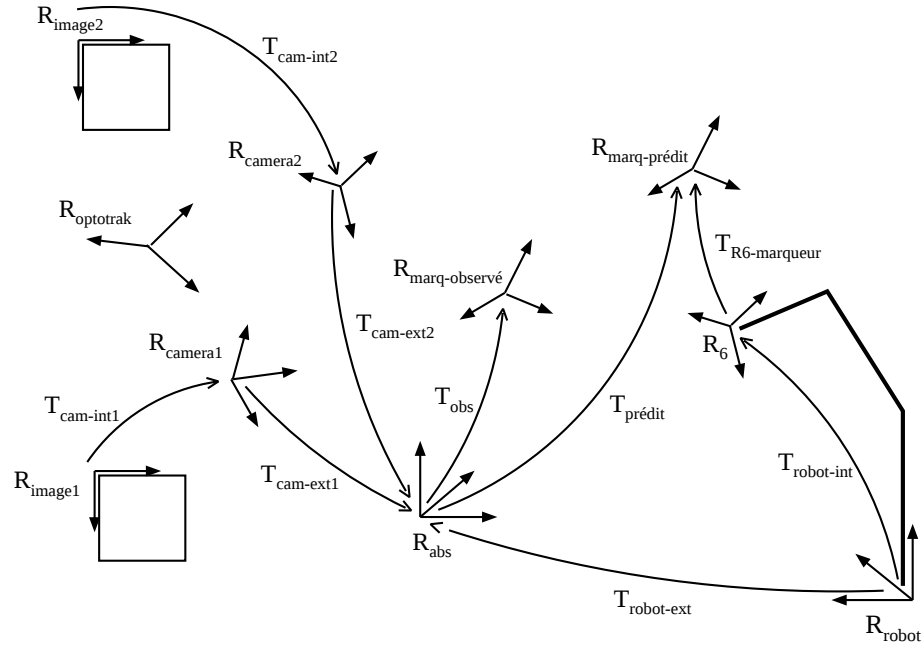


FIG. 7.6 - Les différents repères impliqués dans un cadre général

confondus avec le repère du robot. On se trouve alors dans la situation simplifiée décrite par le schéma 7.7.

Les différentes étapes impliquées sont les suivantes :

- Le **calibrage intrinsèque du robot** a été effectué à l'aide d'optotrak. Nous expliquons succinctement la méthode employée :
 - On fixe à l'extrémité de l'effecteur, un "rigid body" auquel on associe le repère $R_{rigidbody}$.
 - On place le robot dans un grand nombre de configurations différentes (N configurations). Soit $R_{optotrak}$ le repère associé à l'optotrak et dans lequel on fait les mesures. On enregistre pour chaque configuration j , q_j , les coordonnées articulaires du robot et P_j , la position 3D dans $R_{optotrak}$ de l'origine du repère $R_{rigidbody}$ (où de tout autre point fixe dans $R_{rigidbody}$).
 - Les inconnues du système sont :
 - $M_{opto-robot}$, la matrice de passage entre $R_{optotrak}$ et R_{robot} : $(P)_{R_{opto}} = M.(P)_{R_{robot}}$.
 - $P_6 = (X_6, Y_6, Z_6)^t$, la position 3D du point intéressant ((0,0,0) dans $R_{rigidbody}$) dans R_6 , le référentiel associé au dernier segment du robot, sur lequel est fixé le "rigid body".
 - V , le vecteur paramètres du changeur de coordonnées. On notera $C(V, q)$, la matrice de passage entre R_{robot} et R_6 , pour les paramètres V et pour la configuration q .

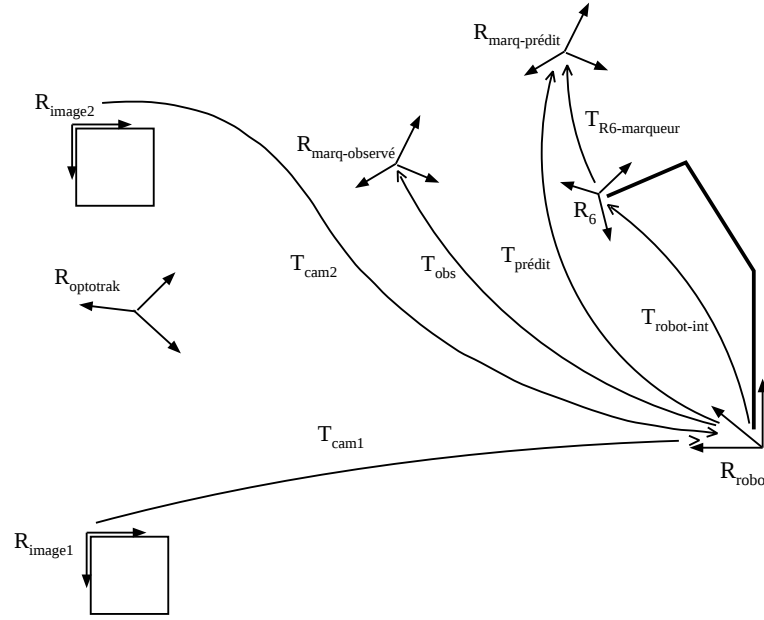


FIG. 7.7 - Les différents repères impliqués dans le cadre de notre expérimentation

- On détermine les inconnues du système (y compris les paramètres du changeur de coordonnées qui sont les seules inconnues qui nous intéressent ici) par une minimisation non linéaire selon la méthode de Levenberg-Marquardt ([PFTV92] p.542-547). On cherche à minimiser la valeur S :

$$S = \sum_{j=1}^N \| (P_j)_{R_{optotrak}} - M_{opto-robot} \cdot C(V, q_j) \cdot P_6 \|^2$$

- Le **calibrage du marqueur** nécessite que le calibrage intrinsèque du robot ait déjà été effectué. Le marqueur est placé sur l'effecteur du robot, c'est-à-dire qu'il est fixe par rapport à R_6 . Un "rigid body" optotrak est également placé sur l'effecteur du robot. Nous avons décomposé le calibrage du marqueur en 2 phases en utilisant le "rigid body" comme intermédiaire :

- **Phase 1** : Détermination de X , la matrice de passage entre R_6 et $R_{rigidbody}$, le repère associé au "rigid body" optotrak.

Cette phase est résolue selon la méthode décrite au chapitre 5.

- **Phase 2** : Détermination de la position 3D des 3 points du marqueur, c'est-à-dire des centres de chacune des croix, dans le repère $R_{rigidbody}$.

On a recours à un second "rigid body" muni d'une pointe. Soit $R_{rigidbody2}$, le repère associé à ce deuxième "rigid body". La position de l'extrémité de la pointe par rapport à $R_{rigidbody2}$ est déterminée en calculant aux moindres carrés le point qui reste fixe dans $R_{optotrak}$, lorsque l'on fait pivoter le "rigid body" autour d'elle.

Une fois cette position connue dans $R_{rigidbody2}$, la pointe nous permet de désigner le centre de chacune des croix et optotrak nous fournit directement la position du point désigné dans le repère $R_{rigidbody}$ (voir photo 5.4 au paragraphe 5.1.3). Une moyenne est effectuée sur une dizaine de saisies pour améliorer la précision.

La combinaison de ces 2 phases fournit la position 3D des 3 points du marqueur dans le repère R_6 .

- La **prédiction de la position 3D du marqueur** est obtenue par la combinaison des étapes de calibrage intrinsèque du robot et de calibrage du marqueur. Sur le schéma 7.7, elle est représentée par la matrice $T_{predict}$.
- Le **calibrage des caméras** est effectué selon la méthode décrite au paragraphe 4.2.2 : la mire de calibrage est portée par le robot et le calibrage est effectué directement dans le repère du robot, ce qui supprime la nécessité d'un recalage.
- La **mesure de la position 3D du marqueur** consiste à déterminer T_{obs} au moyen de la vision. Pour chaque croix, on effectue les étapes suivantes :
 - La position 3D prédite de la croix et les matrices de projection déterminées lors du calibrage des caméras, permettent d'estimer sa position approximative dans chacune des images.
 - Ces positions 2D sont alors affinées grâce à l'algorithme de détection de croix décrit au paragraphe 6.2.4.2.
 - Puis, grâce au calibrage des caméras, on calcule les droites de rétroprojection associées aux positions 2D corrigées dans chacune des deux images.
 - Enfin, la position 3D du centre de la croix est estimée comme l'intersection, au sens des moindres carrés, des deux droites de rétroprojection ainsi obtenues.

7.2.3 Résultats

Pour les expérimentations, le cube portant le marqueur et le “rigid body” optotrak est fixé sur le robot.

7.2.3.1 Expérimentation simple

Etape 1.1 : On place le robot dans une configuration telle que le marqueur est visible par les 2 caméras.

Etape 1.2 : On récupère les coordonnées articulaires $(q_i)_{i=1..6}$, ce qui, grâce au calibrage intrinsèque du robot, nous fournit la matrice de passage entre le repère R_{robot} et le repère R_6 . Soit M , cette matrice.

Chapitre 7. Expérimentations

Etape 1.3 : On connaît les 3 points qui constituent le marqueur : $(P_1)_{R_6}, (P_2)_{R_6}, (P_3)_{R_6}$ par calibrage du marqueur.

Etape 1.4 : Donc on connaît $(P_i)_{R_{robot}} = M.(P_i)_{R_6}$ pour $i = 1..3$.

Etape 1.5 : On projette $(P_i)_{R_{robot}}, i = 1..3$ dans chacune des images. On obtient les points 2D p_{1i} et p_{2i} .

Etape 1.6 : On détecte précisément le centre de la croix dans les images, autour de p_{1i} et p_{2i} . On obtient les points p'_{1i} et p'_{2i} .

Etape 1.7 : On recalcule la position 3D du point dans R_{robot} par :

- calcul des droites de rétro-projection de p'_{1i} et p'_{2i} ,
- calcul du point 3D le plus proche des 2 droites de rétro-projection.

Les résultats affichés sont :

- pour chaque point du marqueur, sa position 3D prévue $(P_i)_{R_{robot}}$ (ce calcul correspond aux étapes 1.1 à 1.4)
- pour chaque point du marqueur, sa position 3D recalculée (étapes 1.1 à 1.7).

On constate des différences de l'ordre du millimètre.

7.2.3.2 Expérimentation avec 2 positions

La transformation entre les repères associés à l'optotrak et au robot n'est pas connue car nous n'avons pas enchainé toutes les étapes de calibrage. Ainsi nous n'avons pas conservé la relation entre ces deux repères calculée lors du calibrage du robot. On ne peut donc pas comparer les mesures absolues effectuées d'une part par l'optotrak dans le repère $R_{optotrak}$, d'autre part, par le système de vision dans le repère R_{robot} . C'est pourquoi nous utilisons deux positions pour effectuer des mesures relatives et ainsi permettre une comparaison avec les résultats fournis par l'optotrak.

Etape 2.1 : On place le robot dans une première configuration.

Etape 2.2 : On effectue les étapes 1.1 à 1.7 de l'expérimentation simple.

Etape 2.3 : On enregistre pour chaque point :

- sa position selon optotrak
- sa position prévue (étapes 1.1 à 1.4)
- sa position recalculée (étapes 1.1 à 1.7).

Etape 2.4 : On déplace le robot dans une deuxième configuration.

Etape 2.5 : Idem étape 2.2.

Etape 2.6 : Idem étape 2.3.

Etape 2.7 : Pour chaque point, on calcule les différences entre :

- la position 1 et la position 2 selon optotrak,
- la position 1 prévue et la position 2 prévue,
- la position 1 recalculée et la position 2 recalculée.

Les résultats affichés sont ceux calculés à l'étape 2.7. Là encore, on constate des différences de l'ordre du millimètre.

7.2.3.3 Simulation d'erreur

Etape 3.1 : On place le robot dans une configuration.

Etape 3.2 : On acquiert les images (que l'on nommera "vraies images").

Etape 3.3 : On acquiert les coordonnées articulaires du robot (que l'on nommera "vraie position").

Etape 3.4 : On simule une petite erreur en translatant le repère R_6 (associé à l'effecteur du robot) d'environ 1 *cm*.

Etape 3.5 : On acquiert les coordonnées articulaires du robot (que l'on nommera "fausse position").

Etape 3.6 : On calcule la translation effectuée entre la "vraie position" et la "fausse position". Cette translation nous servira de référence.

Etape 3.7 : On effectue les étapes 1.4 à 1.7 de l'expérimentation simple avec comme données les "vraies images" et la "fausse position".

L'étape 1.4 fournit la position 3D prévue de chaque point du marqueur, selon la "fausse position" du robot. On nommera cette position 3D "fausse position prévue".

L'étape 1.7 fournit la position 3D de chaque point du marqueur, recalculée grâce à la "fausse position prévue" et aux "vraies images". On nommera cette position "Position Corrigée".

On calcule la translation entre ces 2 positions.

Les résultats affichés sont :

- la "fausse position prévue"
- la "position corrigée"
- la translation de référence

Chapitre 7. Expérimentations

- la translation entre la "fausse position prévue" et la "position corrigée".

On constate que la différence entre la translation de référence et la translation entre la "fausse position prévue" et la "position corrigée" est de l'ordre du millimètre.

De plus, on vérifie visuellement que les projections dans les images de la position corrigée sont bien sur les croix du marqueur.

7.2.3.4 Simulation d'erreur avec 2 positions

Dans cette partie de l'expérimentation, on considère le robot dans 2 configurations différentes. On estime qu'il n'y a pas d'erreur dans la première configuration. En revanche, on simule une petite erreur dans la deuxième configuration.

Etape 4.1 : On place le robot dans une première configuration.

Etape 4.2 : On effectue les étapes 1.1 à 1.7 de l'expérimentation simple.

Etape 4.3 : On enregistre pour chaque point :

- sa position selon optotrak,
- sa position prévue (étapes 1.1 à 1.4),
- sa position recalculée (étapes 1.1 à 1.7).

Etape 4.4 : On déplace le robot dans une deuxième configuration.

Etape 4.5 : On acquiert les images que l'on nommera "vraies images".

Etape 4.6 : On acquiert les coordonnées articulaires du robot que l'on nommera "vraie position".

Etape 4.7 : On acquiert pour chaque point, sa vraie position selon optotrak.

Etape 4.8 : On simule une petite erreur en translatant le repère R_6 du robot, d'environ 1 *cm*.

Etape 4.9 : On acquiert les coordonnées articulaires du robot que l'on nommera "fausse position".

Etape 4.10 : On effectue les étapes 1.4 à 1.7 de l'expérimentation simple avec comme données les "vraies images" et la "fausse position".

L'étape 1.4 fournit la position 3D prévue de chaque point du marqueur, selon la "fausse position" du robot. On nommera cette position 3D "fausse position prévue".

L'étape 1.7 fournit la position 3D de chaque point du marqueur, recalculée grâce à la "fausse position prévue" et aux "vraies images". On nommera cette position "position corrigée".

Étape 4.11 : On calcule pour chaque point, la translation entre la première et la deuxième configurations :

1. selon optotrak (mesures effectuées sur la vraie position),
2. selon les étapes 1.4 de la démo simple, c'est-à-dire selon la prévision de la position des points.
NB : pour la deuxième configuration, ces prévisions ont été effectuées avec la "fausse position".
3. selon la correction par traitement d'images et rétro-projection, c'est-à-dire en utilisant les "positions corrigées" fournies à l'étape 1.7.

Les résultats affichés sont les translations calculées à l'étape 4.11.

On constate que la translation 2. est très différente de la translation de référence obtenue par optotrak, ce qui est normal, puisque son calcul a été effectué avec la "fausse position" pour la deuxième configuration.

En revanche, la translation 3. est peu différente de la mesure de référence (différence de l'ordre de 1 *mm*), ce qui prouve que la correction à partir des images est relativement bonne.

7.3 Conclusion

Ces expérimentations ont permis d'assembler les différentes étapes impliquées par le système de contrôle redondant que nous avons conçu et ont démontré la validité de l'approche.

Les précisions observées sont de l'ordre de 3 *mm* dans un champ d'environ $500 \times 400 \times 400 \text{ mm}^3$ (cas de l'application nucléaire) et de l'ordre du millimètre dans un champ d'environ $300 \times 200 \times 200 \text{ mm}^3$ (cas de l'application médicale).

Ces précisions ont été évaluées en effectuant de la reconstruction 3D à partir de deux caméras observant un même marqueur. Cette manière d'opérer est plus restrictive que celle que nous envisageons qui consiste à utiliser plusieurs caméras observant chacune l'un des marqueurs, pas forcément le même, fixés sur l'effecteur du robot. Néanmoins, on a pu évaluer la précision envisageable avec un tel système de surveillance.

Chapitre 8

Limites et extensions possibles

Les expérimentations décrites au chapitre précédent ont permis de montrer la faisabilité du projet. Cependant, ces expérimentations ont été effectuées dans des “conditions de laboratoire”. Réaliser un contrôle redondant dans des conditions réelles présente des spécificités propres à chaque application. C’est ce que nous détaillerons dans le premier paragraphe.

D’autre part, les résultats précédents ont été obtenus en surveillant un seul marqueur au moyen de deux caméras. Un contrôle complet devra bien sûr intégrer des résultats en provenance de plusieurs caméras surveillant différents marqueurs. Le second paragraphe décrira une architecture possible du système complet, et les fonctionnalités que devraient remplir les procédures de surveillance de haut niveau.

Enfin le dernier paragraphe mettra en exergue d’autres extensions envisageables par rapport à l’avancement actuel du projet.

8.1 Spécificités propres à chaque application

Ces spécificités sont conditionnées par les exigences particulières que requiert chaque application. Nous allons étudier le cas des deux applications qui nous concernent (contrôle redondant dans une salle d’opération et dans un bol de générateur de vapeur d’une centrale nucléaire), mais toute autre application nécessitera une réflexion propre pour l’adaptation du projet.

Les principaux points de difficulté sont traités dans la suite. Il s’agit du choix d’un repère absolu, du choix de la disposition des caméras destinées à la surveillance et de leurs caractéristiques, et enfin du choix des marqueurs à surveiller.

8.1.1 Recalage par rapport à un repère absolu pertinent

Le choix du repère absolu dépend en grande partie de l’objectif que doit réaliser le robot impliqué.

8.1.1.1 Application médicale

L'objectif des Gestes Médico-Chirurgicaux Assistés par Ordinateur (GMCAO) est de fournir au chirurgien des indications pour positionner ses outils par rapport au patient.

Pour le projet de contrôle redondant de la position du système de guidage (en l'occurrence un robot actif ou, ce qui est beaucoup mieux, un robot semi-actif de type Robot à Sécurité Passive [DT95]), il semblerait donc logique de choisir comme repère absolu, un repère lié au patient.

Cependant, de nombreuses applications de GMCAO impliquent de modifier la stratégie opératoire en fonction des mouvements du patient, engendrés par exemple par la respiration.

Différentes phases de calibrage de notre projet s'effectuant par rapport au repère absolu (calibrages extrinsèques du robot et des caméras), nous avons préféré choisir comme référentiel absolu, un repère fixe dans lequel le calibrage extrinsèque du robot et des caméras peut être effectué une fois pour toutes.

D'autre part, la phase d'adaptation de la stratégie opératoire en fonction des mouvements du patient fait partie intégrante de la méthodologie des GMCAO. Elle n'a donc pas à intervenir dans le contrôle redondant du robot, qui doit garder une indépendance la plus grande possible par rapport à ses domaines d'application.

Notons que la détermination de la matrice de passage entre l'un des repères associés au patient (repère associé à des images pré-opératoires ou per-opératoires) et le repère associé au robot fait également partie de la méthodologie des GMCAO (voir [LCDB91] pour un exemple).

Dans la méthode de recalage de caméra décrite au paragraphe 4.2.2, le repère absolu que nous avons choisi est le repère associé au robot. On pourrait choisir un repère différent dans lequel serait connue la position de plusieurs indices visuels. Le recalage des caméras pourrait alors s'effectuer d'une manière similaire à celle employée pour l'application nucléaire. En revanche, cette manière de procéder nécessiterait une étape supplémentaire : le calibrage extrinsèque du robot.

8.1.1.2 Application nucléaire

Le robot est utilisé dans le bol de générateur de vapeur d'une centrale nucléaire pour des tâches d'inspection. Il s'agit d'introduire une caméra dans chacun des tubes d'un échangeur afin d'en vérifier l'état. Le robot est placé dans le bol et les tubes débouchent dans le plafond du bol selon un quadrillage régulier connu.

Pour le projet de contrôle redondant de la position du robot, il est donc logique de choisir comme repère absolu un repère lié au plafond du bol. C'est ce que nous avons fait. D'ailleurs les expérimentations ont été réalisées dans un environnement qui simulait le bol, la mire de calibrage constituant un "faux-plafond".

Une fois ce repère absolu choisi, il faut mettre en place des procédures pour effectuer le calibrage extrinsèque du robot et des caméras.

Pour ce qui est du calibrage extrinsèque des caméras, on peut procéder de la même manière que lors des expérimentations. La difficulté réside ici dans le traitement d'images nécessaire pour détecter l'embouchure des tubes de l'échangeur, ce traitement étant beaucoup plus délicat que celui utilisé pour détecter les trous de la mire (on peut aisément imaginer la différence de difficulté en comparant une image de la mire à une image du plafond du bol, figure 8.1). Une étude préliminaire [Amo93] a cependant montré qu'un tel traitement était envisageable.

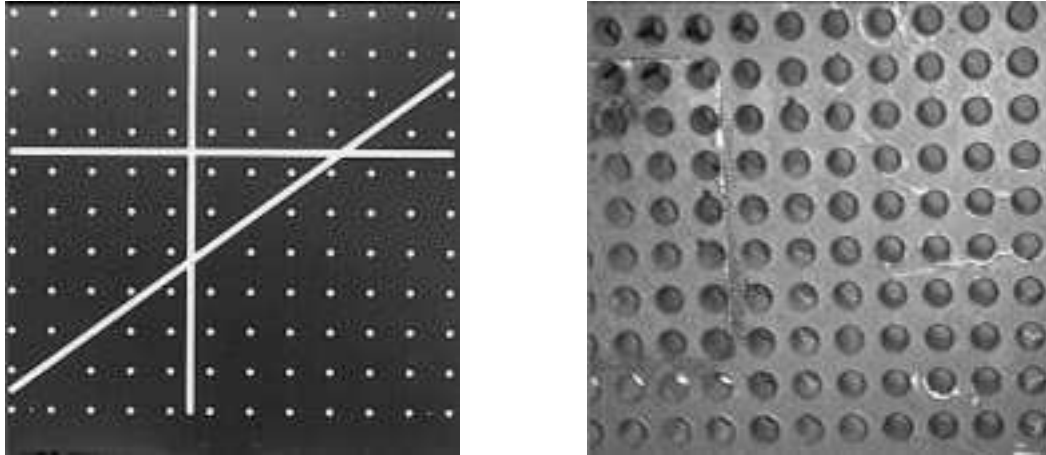


FIG. 8.1 - *A gauche, une image de la mire de calibrage; à droite une image du plafond du bol de générateur de vapeur*

Pour ce qui est du calibrage extrinsèque du robot, on a proposé dans [Poy94] (pages 23-25) une méthode qui utilise toutes les autres phases préliminaires (calibrage intrinsèque du robot, calibrages intrinsèque et extrinsèque des caméras, calibrage des marqueurs) et qui exploite le fait que le robot ne peut être positionné qu'en un certain nombre de positions discrètes pour affiner le résultat.

Les autres choix dépendant des applications, concernent le matériel à employer, à savoir les caméras et les marqueurs.

8.1.2 Choix concernant les caméras

Le choix de la disposition des caméras dépend très fortement de la mission du robot et de l'environnement dans lequel il évolue, cette disposition devant permettre de suivre le robot dans tout l'espace auquel il est susceptible d'accéder (on s'attachera tout particulièrement à suivre l'extrémité du robot, voire uniquement l'outil qu'il porte). La conjugaison des champs de chacune des caméras doit donc couvrir tout cet espace. De plus, les occultations susceptibles de se produire de manière intermittente doivent être minimisées.

Enfin le nombre de caméras et leurs caractéristiques¹ dépendent de la taille de

1. c'est-à-dire principalement l'objectif, la distance de mise au point et la taille de la numérisation

Chapitre 8. Limites et extensions possibles

l'espace à surveiller, de la précision escomptée, mais également du nombre et de la disposition des marqueurs présents.

8.1.2.1 Application médicale

Certaines applications de GMCAO utilisent un robot qui parcourt un espace de travail important. C'est le cas du robot de radiothérapie décrit dans [ASTL94] qui a pour but de positionner un accélérateur linéaire en divers emplacements autour du patient, afin de mieux répartir la dose administrée. Les dimensions de cette application sont très proches de celles de l'application nucléaire.

En revanche, pour les applications de GMCAO plus classiques, qui impliquent un acte chirurgical, l'espace à surveiller est généralement très restreint : un cube d'une trentaine de centimètres de côté est un maximum. La principale difficulté provient donc des occultations possibles dues à la présence d'un personnel hospitalier nombreux. Cette difficulté peut être surmontée en employant un plus grand nombre de caméras et de marqueurs, ce qui augmente la redondance du système. On peut déjà obtenir une bonne redondance avec six marqueurs et quatre caméras disposées au plafond (pour s'affranchir au maximum des occultations engendrées par le personnel) aux quatre coins d'un carré, avec une vue plongeante sur le champ opératoire ; on verra au paragraphe 8.1.3 qu'il est également primordial d'optimiser le placement des marqueurs à l'extrémité du guide robotique.

L'espace à surveiller étant restreint, on peut choisir des caméras ayant des caractéristiques simples :

- Une distance de mise au point fixe est suffisante, ce qui permet de calibrer les caméras comme nous l'avons expliqué au chapitre 3.
- On peut placer les caméras à environ 1.5 mètre de l'espace à surveiller. Une longueur focale d'environ 25 mm et une plaque sensible de taille $4.8 \times 6.4 \text{ mm}^2$ (ce qui constitue des valeurs standard) permettent alors d'englober cet espace.
- Plus la taille de la numérisation sera grande, meilleure sera la précision. Avec les valeurs précédentes, une numérisation de 512×512 pixels implique que chaque pixel représente une surface réelle de $0.6 \times 0.8 \text{ mm}^2$, ce qui constitue déjà une précision honorable.

8.1.2.2 Application nucléaire

Dans cette application, les risques d'occultation sont moindres que dans une application de GMCAO. On redoute seulement des occultations dues à la présence de fils, et au robot lui-même. En revanche, la difficulté provient de la taille de l'espace à surveiller, d'autant plus que l'on est contraint de placer les caméras à l'intérieur du bol.

Les caractéristiques des caméras retenues ne sont donc pas aussi simples que celles des applications de GMCAO. On pourra se reporter à [Poy94] (pages 29-44)

pour une étude détaillée des besoins. On ne présente ici que les conclusions de ce rapport.

Mise au point

Une distance de mise au point fixe est insuffisante, vue la profondeur du champ à observer. Il nous faut en effet avoir des images nettes de 0 à plus de 3 mètres. Une solution simple consiste à effectuer un calibrage intrinsèque pour un ensemble discret de valeurs de mise au point. Au cours du suivi, on choisit la valeur de mise au point en fonction de l'emplacement 3D prévu du marqueur à observer.

La profondeur de champ est proportionnelle au carré de la distance de mise au point et inversement proportionnelle à la distance focale et à l'ouverture [Per91]. Ces considérations nous incitent à choisir une faible ouverture (ce qui implique que l'éclairage ambiant soit important) et une faible focale afin d'avoir une plage de netteté la plus grande possible. Le non respect de ces choix nous obligerait simplement à augmenter le nombre de distances de mise au point, et donc à calibrer les caméras pour ces différentes valeurs.

Une condition sine qua non pour procéder de cette manière est bien sûr une bonne répétabilité de la mise au point.

Enfin, notons que l'on pourrait procéder à un calibrage continu d'une manière similaire à celle décrite en 3.1.4 pour un calibrage de caméra à focale variable. Une adaptation serait néanmoins nécessaire du fait que les deux plans de calibrage doivent se déplacer pour se trouver de part et d'autre du champ de netteté. Ainsi, la variable z , au lieu d'être fixe pour chaque plan, serait une fonction de la valeur de mise au point, les variables x et y de chaque plan étant des fonctions splines de u , v et de la mise au point.

Taille de numérisation

La taille de numérisation choisie dépend de la focale utilisée. La combinaison des deux doit être telle que pour un objet situé à 3.6 mètres de la caméra (distance maximale possible dans le bol), un pixel représente une surface inférieure à 1 mm^2 .

Focale

Le choix de la focale est délicat puisqu'on doit à la fois couvrir un large champ, donc choisir une focale faible, et maintenir une bonne précision, donc choisir une focale relativement élevée.

Cependant, calibrer une caméra qui doit observer un champ d'une taille donnée, nécessite une mire d'une taille supérieure ou égale. Dans le cas d'une focale faible, ceci n'est guère envisageable d'un point de vue pratique : pour une focale de 6 mm , et une plaque sensible standard ($4.8 \times 6.4 \text{ mm}^2$), la surface observée à 3 mètres est de $2.4 \times 3.2 \text{ m}^2$ (évaluation faite selon un modèle à sténopé).

Chapitre 8. Limites et extensions possibles

En revanche, si l'on utilise des focales plus grandes, le nombre de caméras nécessaires pour couvrir tout l'espace à observer, devient prohibitif.

Ainsi, le choix de la longueur focale est intimement lié au choix du nombre de caméras.

Nombre de caméras

Des simulations [Poy94] ont montré que même en utilisant une focale de 6 *mm*, il faut au moins une dizaine de caméras pour surveiller la partie supérieure² du bol, située à moins de 1.25 *m* du plafond.

Cette approche étant totalement irréaliste, nous avons alors étudié la possibilité d'utiliser des caméras montées sur une platine mobile disposant de deux degrés de liberté en rotation.

Il y a bien évidemment besoin de moins de caméras si elles sont orientables. On peut même dire qu'une seule caméra est capable d'englober tout le bol du moment que l'amplitude des mouvements de la platine le permet (360° en azimut et 90° en hauteur). Trois caméras seraient donc suffisantes pour assurer la redondance même en cas d'occultations³.

D'autre part, comme la procédure de recalage de caméra est longue, on ne peut pas se permettre de l'exécuter à chaque mouvement de la platine. Cette approche nécessite donc le calibrage de la platine elle-même (qui peut être effectué en dehors du bol) et requiert une procédure de recalage de l'ensemble platine-caméra à la place d'un simple recalage de caméra.

Disposition des caméras

Les caméras doivent être placées de telle sorte qu'elles voient au moins en partie le plafond du bol, afin de pouvoir se recalibrer par rapport au référentiel absolu. Ceci est peu contraignant dans le cas de caméras mobiles ; en revanche, dans le cas de caméras fixes, cette contrainte ajoutée au champ d'observation limité des caméras, crée des "zones mortes", c'est-à-dire des zones impossibles à observer.

En conclusion, le choix de caméras mobiles (3 devraient être suffisantes) avec une focale de 25 *mm* et une numérisation de 1024 × 1024 pixels, semble un bon compromis.

De plus la possibilité de commander la distance de mise au point avec une bonne répétabilité est indispensable. Afin de réduire le nombre de valeurs de mise au point, on choisira de préférence une faible ouverture, ce qui permet d'obtenir une

2. On ne considère que la partie supérieure du bol, car il subsiste des "zones mortes" dans la partie inférieure à cause de la contrainte imposée par le recalage des caméras, à savoir que la caméra doit observer une partie du plafond.

3. Deux caméras permettent une vérification 3D (même si elles n'observent pas le même marqueur) ; la troisième assure la redondance.

profondeur de champ plus importante.

Enfin les caméras seront placées de préférence en bas du bol afin de regarder aisément une partie des trous du plafond lors de la phase de recalage.

8.1.3 Choix concernant les marqueurs

Contrairement aux choix concernant les caméras, les choix concernant les marqueurs ne sont pas fondamentalement différents d'une application à l'autre. Nous allons donc étudier chaque caractéristique de manière générale avant de spécifier les solutions retenues pour chaque application.

Les différents choix concernant les marqueurs portent sur leur forme, leur taille, leur matériau constitutif, leur nombre et leur disposition sur la partie du robot à surveiller (essentiellement le porteur de l'outil).

Forme

La forme des marqueurs a déjà été étudiée au chapitre 6.

Taille

On a énoncé des conditions sur l'apparence (en particulier la taille) des marqueurs dans les images, permettant d'assurer une bonne détection. Ces conditions 2D sur les images des marqueurs doivent être transcrites en conditions 3D sur les marqueurs eux-mêmes. La taille des marqueurs dépend donc des caractéristiques des caméras employées (focale, taille de numérisation) et de la distance maximale susceptible de séparer le marqueur d'une caméra.

Soient $s \times s \text{ mm}^2$, la surface réelle observée par la caméra à une distance d et $N \times N$ pixels la taille de la numérisation. Supposons que le marqueur est observé selon une incidence inférieure à θ_{max} . Alors une condition 2D qui imposerait qu'un segment ait une longueur d'au moins n pixels, se traduit par une condition 3D qui impose que le segment correspondant sur le marqueur réel ait une longueur d'au moins $l \text{ mm}$, avec $l = \frac{n.s}{N.\cos(\theta_{max})}$. Dans le cas de marqueurs constitués de croix, cette formule peut être utilisée pour traduire des conditions sur la longueur et la largeur des branches des croix. A titre indicatif, voici deux applications numériques grossières :

- $s \approx 250 \text{ mm}$, $N = 512$ et $\alpha = 45^\circ$ impliquent que $l \approx 0.7n$. Ces valeurs correspondraient à une application du type de celles des GMCAO. Ainsi, pour que l'on voit une croix dont les branches ont une longueur d'au moins 10 pixels et une épaisseur d'au moins 2 pixels, il faudrait que les croix réelles aient des branches d'une longueur supérieure à 7 mm et d'une épaisseur supérieure à 1.5 mm .

- $s \approx 500 \text{ mm}$, $N = 1024$ et $\alpha = 75^\circ$ impliquent que $l \approx 1.9n$. Ces valeurs correspondraient à une application du type de celle de l'EDF, avec des caméras mobiles. Ainsi, pour que l'on voit une croix dont les branches ont une longueur d'au moins 10 pixels et une épaisseur d'au moins 2 pixels, il faudrait que les croix réelles aient des branches d'une longueur supérieure à 19 mm et d'une épaisseur supérieure à 4 mm .

Nombre et disposition

Les marqueurs sont regroupés autour d'une région restreinte à surveiller (un repère est associé à cette région) : il s'agit essentiellement de l'extrémité du robot (repère R_6 dans le cas d'un robot à 6 degrés de liberté) ou de l'outil qu'il porte (repère R_{outil}) ; mais il peut également s'agir de chacune des articulations du robot (repère R_i) si l'on veut effectuer une vérification axe par axe.

La région sera considérée comme bien surveillée si, à tout instant, quelle que soit la pose⁴ du repère associé à cette région, deux des caméras peuvent observer l'un des marqueurs du groupe (pas forcément le même) selon une incidence inférieure à θ_{max} . La majoration de l'incidence permet de limiter les effets de perspective et donc de faciliter le traitement d'images pour une détection précise de la position du marqueur.

Notons que si une seule caméra est en mesure d'observer le groupe de marqueurs, la mesure de la distance entre la position prévue du marqueur et la droite de rétroprojection associée à son image ne fournit pas de vérification en profondeur, c'est-à-dire dans la direction caméra-marqueur. En revanche, on a une bonne vérification selon des directions perpendiculaires.

Dans le cas où une seule caméra est valide, on pourrait utiliser une méthode directe de détermination de la pose du marqueur dans le repère caméra, telle que celles décrites dans le paragraphe 4.1.2.3 pour effectuer le calibrage extrinsèque des caméras. Cependant, ces méthodes directes sont sensibles au bruit et ne permettraient donc pas d'obtenir une position précise. Néanmoins l'information en profondeur ne serait pas complètement inexistante.

On pourrait donc envisager d'utiliser la procédure la plus simple de surveillance (c'est-à-dire mesurer la distance entre la position prévue du marqueur et la droite de rétroprojection de son image) tant qu'au moins deux caméras sont en mesure d'observer l'un des marqueurs selon une incidence inférieure à θ_{max} , puis de changer de stratégie, c'est-à-dire déterminer explicitement la pose du marqueur, dès qu'il n'y plus qu'une seule caméra valide.

Au lieu d'imposer que seulement deux caméras puissent observer l'un des marqueurs, on peut imposer que trois caméras, au moins, puissent observer l'un des marqueurs du groupe (pas forcément le même) selon une incidence inférieure à θ_{max} .

4. pose = position et orientation

Ceci permet d'envisager une éventuelle occultation du groupe de marqueurs pour l'une des caméras.

Le nombre et la disposition des marqueurs sont intimement liées au nombre et à la disposition des caméras, puisqu'un nombre plus important de marqueurs permet qu'au moins l'un d'entre eux soit vu par un nombre plus restreint de caméras. Or il est moins coûteux d'augmenter le nombre de marqueurs plutôt que le nombre de caméras. C'est donc ce que nous ferons dans la mesure du possible.

Afin d'optimiser la disposition des marqueurs, nous les placerons sur les faces d'un polyèdre régulier convexe, ce qui permettra de les orienter selon des directions régulièrement réparties.

Pour une application de GMCAO dans laquelle le robot restera à peu près dans la même position et orientation, et dans laquelle les caméras sont fixées au plafond, on peut envisager de disposer 6 marqueurs sur la "calotte supérieure" d'un dodécaèdre régulier, ce qui implique que les normales à deux faces contiguës soient séparées d'un angle d'environ 63° (voir figure 8.2). On peut bien sûr choisir toute autre forme géométrique convexe (icosaèdre, octaèdre tronqué, icosaèdre tronqué ... [Wen71]).

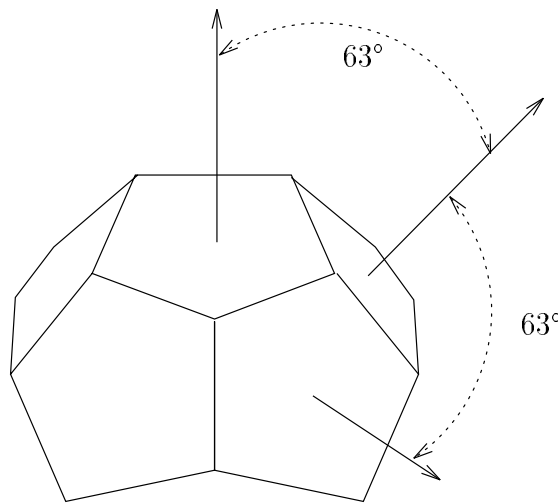


FIG. 8.2 - Dans le cadre d'une application où le robot bouge peu, on peut disposer les marqueurs sur les faces d'un dodécaèdre, lui même placé au dos de l'outil à surveiller. Leur normales ont alors des directions qui diffèrent de 63° .

Ce groupe solidaire de marqueurs est placé au dos de l'outil.

Pour l'application nucléaire, on peut envisager le même genre de disposition pour la surveillance de l'extrémité du robot à proximité du plafond. En effet, par rapport aux applications de GMCAO, la géométrie de la scène est simplement renversée : les caméras sont placées en bas de la scène ; le robot travaille à proximité du plafond

Chapitre 8. Limites et extensions possibles

avec pour objectif d'introduire un outil dans l'un des tubes du plafond, c'est-à-dire selon une orientation constante.

En revanche, dans les deux types d'applications, il est plus délicat d'optimiser la disposition des marqueurs pour surveiller le robot loin de son objectif ou bien pour surveiller l'une de ses articulations antérieures. En effet, dans ces cas-là, on n'a pas d'a priori sur la position et l'orientation de la partie du robot à surveiller. Il faudrait donc instrumenter le robot avec un polyèdre entier de marqueurs, de telle sorte que, quelle que soit l'orientation du segment concerné, au moins un marqueur puisse être vu par chaque caméra selon un angle d'incidence inférieur à θ_{max} .

Or il n'est matériellement pas possible d'instrumenter ainsi un segment du robot près d'une articulation⁵. On a donc étudié une solution intermédiaire qui consiste à entourer le segment du robot de marqueurs.

L. Carrat a effectué des simulations dans le cas du bol de générateur de vapeur d'une centrale nucléaire [Car94]. Les hypothèses ont été les suivantes : on veut surveiller un segment modélisé par un prisme à base carrée, dont les quatre côtés ont été instrumentés avec un marqueur ; on cherche à positionner n caméras mobiles sur le pourtour du bol, de façon que, quelle que soient la position et l'orientation du prisme dans le bol, au moins deux d'entre elles puissent observer l'un des marqueurs selon une incidence inférieure à θ_{max} ; les caméras doivent respecter des contraintes d'éloignement les unes des autres, afin de ne pas observer la scène sous des angles très proches (en l'occurrence, on a imposé qu'elles soient séparées d'au moins 50 cm). Le programme de simulation discrétise les positions et orientations possibles pour le prisme, ainsi que les positions des n caméras sur le pourtour du bol.

Pour $n = 3$ caméras, on a trouvé une configuration des caméras telle qu'au moins deux d'entre elles voient un marqueur selon une incidence inférieure à 82° . C'est le meilleur angle que nous ayons obtenu (c'est-à-dire le plus petit). Cette valeur très élevée risque de déformer les marqueurs de manière significative, accentuant ainsi la difficulté à les détecter.

Pour $n = 4$ caméras, le meilleur angle d'incidence maximum trouvé vaut 75° .

Ces résultats peu satisfaisants pourraient être améliorés en augmentant à la fois le nombre de caméras et le nombre de marqueurs (par exemple en disposant 6 marqueurs sur les faces d'un prisme à base hexagonale, ou bien même en multipliant par 2 le nombre de marqueurs comme sur la figure 8.3).

Cette disposition permet d'améliorer la couverture des directions, mais elle crée un relief le long des segments du robot, ce qui modifie son modèle CAO et réduit son champ d'action afin d'éviter des collisions avec les autres segments du robot. Ainsi, si l'on ne veut pas créer un relief trop important le long des segments du robot, la direction d'observation parallèle à l'axe du segment, sera toujours pénalisée.

5. En fait, il serait possible de le faire, mais en déportant le polyèdre couvert de marqueurs à une certaine distance du bras, ce qui induirait des problèmes de collision avec les autres segments du robot.

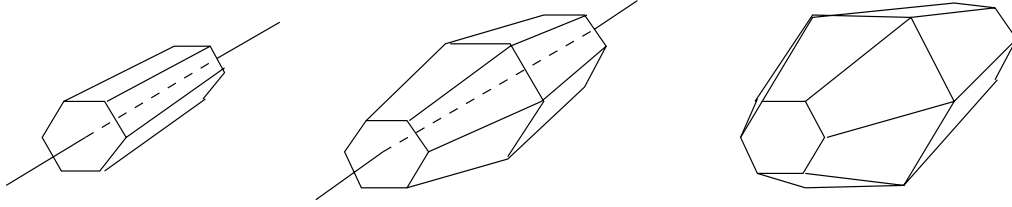


FIG. 8.3 - Les marqueurs peuvent être disposés sur chacune des faces de ces polyèdres (sauf bien sûr les deux faces hexagonales). Le premier est un prisme droit à base hexagonale. On voit clairement que si une caméra l'observe selon une direction presque parallèle à son axe, les marqueurs seront très difficiles à détecter à cause des effets de perspective importants. Le deuxième polyèdre permet d'améliorer la couverture des directions, mais comporte encore une direction d'observation pénalisée. Le troisième améliore davantage la couverture des directions, mais présente un relief important.

Matériau

Le matériau constitutif des marqueurs doit permettre de les détecter facilement et précisément dans les images et, dans la mesure du possible, de les distinguer de l'environnement.

Pour le moment, nous avons retenu des motifs blancs sur un fond noir. Le matériau noir est un simple velours absorbant la lumière. Le matériau blanc est un revêtement rétro réfléchissant utilisé habituellement en prévention routière pour être vu de loin. Ce matériau est constitué de petites billes encapsulées dans un adhésif. Nous en avons testé divers types, commercialisés par la société "Le Marquage Industriel" : revêtements rétro réfléchissants ScotchliteTM Séries 6060, 13150, 2000 X, 7800 et 7610. C'est ce dernier qui s'est avéré le plus performant, pour le bon contraste qu'il fournit.

L'intérêt de ce matériau est qu'il n'est pas exigeant vis-à-vis de la source lumineuse, ni du point de vue de sa nature, ni du point de vue de sa direction. Il suffit qu'il soit observé selon une direction peu différente de la direction d'illumination pour être utilisé de manière optimale. Un spot placé à côté de la caméra, ou mieux, un éclairage réparti autour de l'objectif de la caméra, réalisent cette condition.

Néanmoins, il serait intéressant de s'affranchir au maximum des autres objets de l'environnement susceptibles de réfléchir une quantité importante de lumière. C'est particulièrement le cas dans le domaine des GMCAO où de nombreux outils chirurgicaux sont présents dans le champ des caméras.

Une solution simple consiste à utiliser des marqueurs d'une couleur absente de l'environnement. Si l'on dispose de caméras couleur, la discrimination est relativement simple.

Si ce n'est pas le cas, un dédoublement de l'information lumineuse et la combinai-

son de différents filtres permettrait de sélectionner les points émettant uniquement la longueur d'onde choisie.

8.2 Ce qu'il reste à faire : la gestion de haut niveau

On vient de voir que pour que le contrôle redondant soit efficace, il faut disposer de plusieurs caméras (au moins trois) et de nombreux marqueurs. Dans les chapitres précédents, nous avons étudié la faisabilité du contrôle sur un seul marqueur. Il faut maintenant mettre en place une architecture et des procédures de haut niveau pour gérer tous les marqueurs et toutes les caméras.

8.2.1 Architecture

L'acquisition des images et leur traitement constituent les difficultés majeures rencontrées lors de la vérification en temps réel de la position du robot. Une architecture envisageable pour s'affranchir le plus possible de ce goulot d'étranglement, consiste à décentraliser les traitements concernant les caméras afin de les effectuer simultanément plutôt que consécutivement. Aussi nous proposons une architecture constituée d'un contrôleur central et de plusieurs processeurs décentralisés. Un processeur décentralisé serait associé à chacune des caméras. Le contrôleur central serait chargé de récolter les résultats intermédiaires fournis par les processeurs décentralisés et de les intégrer afin d'effectuer une vérification globale. Des échanges de messages entre ces deux types de composants assureraient la synchronisation des acquisitions, la transmission des requêtes et des résultats intermédiaires ...

Nous allons détailler le rôle de chacun des composants, dans le cas où l'on disposerait de caméras mobiles.

8.2.2 Rôle du contrôleur central

Le contrôleur central est chargé de la gestion de haut niveau : il effectue des requêtes auprès de chacun des processeurs décentralisés et analyse les résultats pour décider si le contrôle est correct ou non. Son travail peut se résumer par l'algorithme suivant :

- il initialise le processus de contrôle redondant :
 - il acquiert les valeurs articulaires courantes du robot,
 - il prédit la position et l'orientation de tous les marqueurs,
 - pour chaque processeur i :
 - il établit la liste des marqueurs visibles par la caméra i dans de bonnes conditions (sous un angle d'incidence inférieur à θ_{max}),

- il optimise la répartition des marqueurs à surveiller par chaque caméra,
 - il envoie une requête au processeur i , c'est-à-dire la liste des marqueurs à surveiller et leur position approximative.
- tant qu'aucune erreur n'est détectée :
- s'il n'y a pas de messages en attente, il attend les messages en provenance des processeurs décentralisés, sinon il traite le premier message de la liste d'attente.
 - selon la nature du message, il effectue le traitement approprié :
 - si le processeur i demande la configuration courante du robot, il acquiert les valeurs articulaires courantes du robot et les envoie au processeur i
 - si le processeur i envoie un nouveau résultat :
 - il intègre le nouveau résultat à la liste des résultats récents,
 - il analyse la cohérence des informations les plus récentes,
 - si la cohérence est mauvaise, il signale qu'une erreur a été détectée et déclenche un signal d'alarme.
 - si le processeur i envoie un message de fin d'exécution de la requête précédente, il envoie une nouvelle requête au processeur i :
 - il acquiert les valeurs articulaires courantes du robot,
 - il revoit éventuellement la stratégie globale, c'est-à-dire qu'il peut assigner au processeur i un autre groupe de marqueurs à surveiller (par exemple lorsque l'une des platines orientables est proche de ses butées mécaniques),
 - il envoie au processeur i une nouvelle requête constituée de la liste des marqueurs à surveiller et de leur position approximative ; la liste des marqueurs peut être quasiment la même que précédemment s'il n'y a pas eu de changement de stratégie globale, ou bien totalement différente si la caméra i observe une autre articulation.

8.2.3 Rôle des processeurs décentralisés

Le rôle d'un processeur décentralisé se décompose en deux types d'activité principales : la gestion du système d'acquisition d'image (platine mobile, paramètres de la caméra, acquisition d'image) et le traitement des données (traitement d'images, calcul des droites de rétroprojection ...). Son travail peut se résumer comme une boucle sur l'ensemble des actions suivantes :

- il attend une requête de la part du contrôleur central.

Chapitre 8. Limites et extensions possibles

- dès que la requête arrive, il en extrait la liste du groupe de marqueurs à surveiller et sa position approximative,
- il règle la platine mobile supportant la caméra pour que celle-ci soit orientée en direction du groupe de marqueurs à surveiller,
- il règle la distance de mise au point de la caméra (et éventuellement d'autres paramètres tels que la distance focale, l'ouverture ...),
- pour chaque marqueur j à surveiller :
 - il envoie un message au contrôleur pour lui demander les valeurs courantes des codeurs du robot, et attend la réponse,
 - il acquiert l'image,
 - il essaie de détecter l'image du marqueur j ,
 - si il y parvient,
 - il calcule la droite de rétroprojection associée,
 - il calcule la distance entre la position prévue du marqueur j et la droite de rétroprojection associée à son image,
 - il envoie un message au contrôleur contenant un booléen indiquant que le marqueur j a été détecté et la distance qui vient juste d'être calculée,
 - il met à jour la probabilité d'apparition d'une occlusion (en fonction de l'analyse effectuée lors du traitement d'image) concernant le marqueur j ; ceci permettra ultérieurement de savoir dans quelle mesure on a affaire à une simple occlusion du marqueur ou à une véritable erreur,
 - sinon il envoie un message au contrôleur contenant un booléen indiquant que le marqueur j n'a pas été détecté et la valeur courante de la probabilité d'apparition d'une occlusion du marqueur j ,
- il envoie un message au contrôleur central pour lui signaler qu'il a terminé d'examiner tous les marqueurs.

8.3 Autres extensions possibles

Nous montrons d'abord que le recours à l'Optotrak est pratique, mais n'est pas indispensable ; celui-ci pourraient en effet être remplacé par un système de vision stéréoscopique très simple.

Les autres extensions envisagées ici interviendraient après la détection d'une panne par le système de contrôle redondant. Il s'agit d'une part de diagnostiquer l'origine de la panne, d'autre part d'exploiter les informations fournies par le système de contrôle redondant pour sortir le robot d'une situation critique, malgré la panne.

8.3.1 Suppression de l'Optotrak

Le principal intérêt de remplacer l'Optotrak par un système de stéréovision est bien sûr le moindre coût d'un tel système. C'est ici aisément envisageable car chaque fois que nous avons eu recours à l'Optotrak au cours des diverses phases de calibrage, il a servi à localiser un "rigid body" de manière statique. Or on pourrait substituer au "rigid body" des indices visuels permettant de constituer un repère 3D. Donc tout autre localisateur 3D pourrait remplacer l'Optotrak, en particulier un simple système de stéréovision.

Un tel système pourraient être constitué de deux caméras et d'un ou deux marqueurs du type de ceux conçus pour le projet de contrôle redondant.

Les deux caméras devraient être calibrées par rapport à un même repère. Cela peut se faire de deux manières :

- Première méthode :

Les deux caméras sont d'abord calibrées intrinsèquement, indépendamment l'une de l'autre (par exemple selon la méthode des deux plans, grâce à une mire que l'on peut translater mécaniquement, d'une longueur connue, selon un axe perpendiculaire à son plan).

Puis elles sont recalées par rapport à un repère commun. Ce recalage est plus simple que lors des expérimentations que nous avons décrites au paragraphe 7.1.3, puisque nous pouvons choisir un objet de référence plus adapté.

- Deuxième méthode :

Les deux caméras sont directement calibrées dans un repère commun, par exemple grâce à une mire de calibrage qui autorise mécaniquement des rotations du plan contenant les points de calibrage et qui est capable de les mesurer précisément.

Une fois les deux caméras calibrées dans un même repère, on peut employer le système pour localiser un marqueur dans ce repère.

- Chaque caméra acquiert une image.
- Un algorithme de traitement d'image détecte les points caractéristiques du marqueur dans chacune des images. Cet algorithme pourra être très semblable à celui utilisé dans le projet de contrôle redondant ; cependant, il devra intégrer une procédure de reconnaissance du marqueur, d'une part pour s'assurer qu'il s'agit bien d'un marqueur (en effet, on ne dispose pas ici, d'une phase de prédiction qui permet de réduire la fenêtre de recherche du marqueur ; on est donc amené à traiter l'image dans son intégralité et les sources d'ambiguïté sont plus nombreuses), d'autre part pour distinguer les marqueurs les uns des autres, si l'on veut en employer plusieurs simultanément.

Chapitre 8. Limites et extensions possibles

- On calcule les droites de rétroprojection associées à chaque point caractéristique, et l'on retrouve la position 3D de ce point par intersection des droites de rétroprojection. Il ne s'agit pas d'une véritable intersection, les droites pouvant être disjointes, mais du calcul du point le plus près des deux droites, au sens des moindres carrés.

Une localisation statique précise au moyen d'un système de stéréovision est donc aisée. En revanche, une localisation dynamique tout aussi précise au moyen de simples caméras [Duc95] serait un projet beaucoup plus ambitieux qui trouverait des applications immédiates, notamment dans le domaine des GMCAO, résolvant ainsi des problèmes de coût, des problèmes de stérilisation du matériel (opération délicate sur des rigid bodies Optotrak, qu'on ne peut pas se permettre de jeter après chaque utilisation), et les problèmes liés à la présence de fils électriques à proximité du patient.

Un projet déjà bien avancé dans ce domaine est décrit dans [MH95].

Les deux paragraphes suivants décrivent des extensions du projet de contrôle redondant qui prendraient place après la détection d'un panne.

8.3.2 Diagnostic de l'origine de la panne

Une fois un problème détecté, il est intéressant d'en déterminer la nature et l'origine.

Il peut s'agir d'une dérive, provenant du fait que les paramètres du robot (déterminés lors de son calibrage) ne sont plus valables, ce qui pourrait être le signe d'un petit problème mécanique (usure, jeux ...) ou d'une utilisation non conforme du robot (charge trop importante, utilisation hors de l'espace de travail pour lequel il a été calibré ...).

Ce type de problèmes se reconnaît par des erreurs relativement modérées et est détecté plus ou moins rapidement selon le seuil d'erreur autorisé.

Au contraire, lorsque l'erreur détectée est importante, c'est qu'il s'agit d'un incident matériel. Dans le cas d'un codeur cassé, ou d'un problème de communication de l'une des valeurs codeurs, il est possible de déterminer quel est le codeur concerné, par l'examen successif des groupes de marqueurs placés sur chacun des segments du robot.

Un diagnostic de la panne est intéressant pour décider de la suite des opérations.

Une simple dérive empêche le robot d'effectuer sa tâche correctement, mais ne le rend pas fondamentalement dangereux. On peut donc se permettre de le ramener à sa position initiale en prévoyant malgré tout une bonne marge de sécurité dans ces déplacements. Ceci permettra d'interrompre l'opération en cours, en toute sécurité.

En revanche, si l'on ne peut pas compter sur l'une des valeurs codeurs, les algorithmes classiques de suivi de trajectoire échouent et le retour du robot à sa

position initiale devient périlleux avec uniquement un contrôle de très bas niveau. Or le système de contrôle redondant peut fournir des informations de plus haut niveau. C'est ce que nous allons voir dans le paragraphe suivant.

8.3.3 Commande du robot grâce à la vision

Dans la situation que nous venons de décrire, le retour d'information ne peut plus s'effectuer en lisant la valeur des codeurs. En revanche, la position des marqueurs dans l'espace et dans les images, est directement liée à la configuration du robot. Elle peut donc être exploitée pour estimer les valeurs articulaires du robot. On peut alors effectuer un asservissement articulaire classique en utilisant les valeurs estimées par la vision au lieu d'utiliser les valeurs fournies par les codeurs (paragraphe 8.3.3.1).

La position des marqueurs peut également être exploitée directement dans la boucle d'asservissement du robot. Cette manière d'appréhender le problème s'intègre dans la notion de "commande référencée capteur". Deux approches peuvent alors être retenues : l'asservissement en position qui cherche à réguler une information spatiale (paragraphe 8.3.3.2), et l'asservissement visuel qui cherche à réguler une information 2D (paragraphe 8.3.3.3).

8.3.3.1 Asservissement articulaire

L'envoi du robot à une configuration désirée q^* se fait de manière dynamique selon le schéma de la figure 8.4.

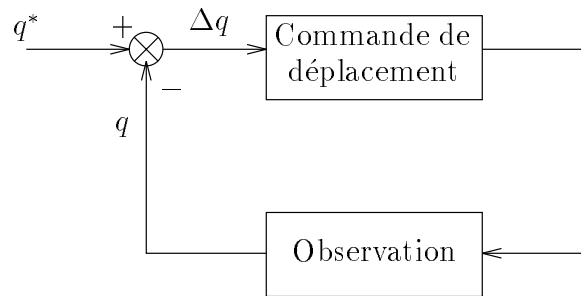


FIG. 8.4 - *Asservissement articulaire*

L'aspect dynamique de l'asservissement consiste à modifier la commande du robot en temps réel, c'est-à-dire dès qu'une nouvelle information sur les valeurs articulaires courantes est disponible, sans attendre que la commande précédente soit exécutée complètement. Ceci permet d'ajuster en permanence la consigne de déplacement et donc d'effectuer celui-ci plus rapidement et plus précisément.

En première approximation⁶, la commande consiste à imposer à chaque moteur i une vitesse proportionnelle à Δq_i : $\dot{q} = k(q^* - q)$.

6. En réalité, il s'agirait plutôt d'un contrôle de type P.I.D. (proportionnel - intégral - dérivé).

Chapitre 8. Limites et extensions possibles

Dans notre cas, les valeurs articulaires courantes ne seraient pas directement fournies par le robot, mais évaluées grâce aux informations visuelles fournies par les marqueurs. Nous allons voir comment peut se faire une telle évaluation.

Dans la suite, on supposera que chaque marqueur est identifiable par les procédures de traitement d'images et que chaque segment du robot est instrumenté avec plusieurs marqueurs. D'autre part on utilisera uniquement la représentation ponctuelle des marqueurs, qui consiste à considérer un marqueur comme un ensemble de points caractéristiques. On n'aura en effet pas besoin d'associer un repère à un marqueur. Enfin, on utilisera le terme "indice" pour désigner l'un des points caractéristiques du marqueur.

Nous allons décomposer la détermination des valeurs articulaires du robot grâce à la vision, en trois étapes :

– Détermination de la position 3D des indices

Le traitement de l'image j permet d'identifier les indices présents dans l'image et de déterminer leurs coordonnées 2D $(u_{s,i,j}, v_{s,i,j})$ (s est le numéro du segment du robot sur lequel est placé l'indice et i est le numéro de l'indice parmi les indices du segment s), puis de calculer dans R_{abs} , la droite de rétroprojection $\mathcal{D}_{s,i,j}$ qui leur est associée.

Si un même indice est vu dans plusieurs images, on peut estimer sa position 3D comme l'intersection aux moindres carrés des diverses droites de rétroprojection qui lui sont associées.

– Détermination de la pose des segments du robot

Pour un segment s donné, on connaît par calibrage, les coordonnées $P_{s,i}$ exprimées dans le repère R_{seg_s} , de tous les indices placés sur ce segment.

Si la première étape a permis de reconstruire dans R_{abs} au moins trois indices appartenant au segment s , alors on peut chercher la transformation T_s qui les mettra en correspondance avec leurs homologues exprimés dans R_{seg_s} , par exemple selon la méthode décrite dans [AHB87]. La transformation T_s ainsi obtenue est la pose du segment s dans le repère R_{abs} .

En revanche si l'on n'a pas pu reconstruire au moins trois indices, ce type de méthodes n'est plus envisageable. On peut cependant estimer la pose T_s en remarquant que chaque indice $P_{s,i}$ se trouve théoriquement sur chacune des droites de rétroprojection $\mathcal{D}_{s,i,j}$ qui sont associées à ses images. La pose T_s de R_{seg_s} dans R_{abs} peut alors être définie comme la matrice de passage qui minimise l'ensemble des distances entre les indices et les droites de rétroprojection qui leur sont associées, ce qui peut se traduire par la minimisation de la fonction suivante :

$$f(T_s) = \sum_{\{P_{s,i}\}} \sum_{\{\mathcal{D}_{s,i,j}\}} d^2(T_s \cdot (P_{s,i})_{R_{seg_s}}, (\mathcal{D}_{s,i,j})_{R_{abs}})$$

On a donc à minimiser le même type de fonction que pour le recalage de caméras (voir équation 4.1). Le problème est cependant mieux conditionné du fait que les droites de rétroprojection ont des directions très différentes (puisqu'elles proviennent de caméras différentes).

Notons qu'une fois T_s déterminée, on peut retrouver la position dans R_{abs} de tous les indices placés sur le segment s . Ceci n'est pas utile ici, mais le sera au paragraphe 8.3.3.2.

– Détermination des valeurs articulaires

La pose T_s de chaque segment s ayant été déterminée, on peut retrouver les valeurs articulaires du robot de manière incrémentale (d'abord q_1 à partir de la pose du premier segment T_1 , puis q_2 à partir de q_1 et T_2 ...). Ceci n'est bien sûr possible que si l'on dispose de suffisamment d'informations.

Dans le cas contraire on peut procéder de manière plus globale en utilisant les changeurs inverses correspondant à des modèles du robot limités à quelques segments successifs du robot. Soit C_{ij} , le changeur partiel permettant de passer de R_{seg_i} à R_{seg_j} , où $i < j$ et R_{seg_0} est assimilé à R_{abs} .

Soit seg_i le premier segment pour lequel on a suffisamment d'informations. Alors on détermine les i premières valeurs articulaires grâce à "l'inverse de C_{0i} " et à T_i . Pour $i > 1$, il est probable qu'on obtienne des configurations multiples. Dans ce cas, les informations partielles fournies par la vision permettent de déterminer la configuration effective.

Soit seg_j le deuxième segment pour lequel on a suffisamment d'informations. Alors on détermine les $j - i$ valeurs articulaires suivantes grâce à C_{0i} instancié avec les valeurs déterminées précédemment, à "l'inverse de C_{ij} " et à T_j . On continue ainsi jusqu'au dernier segment S .

On peut également chercher à estimer les valeurs articulaires du robot dès la phase de minimisation sans passer par la détermination explicite de la pose de chaque segment. Pour cela, on modifie la fonction à minimiser en considérant les valeurs articulaires $q = (q_1, \dots, q_s)$, comme les inconnues à identifier. Soit $C_{0s}(q_1, \dots, q_s)$ le changeur partiel du robot jusqu'au segment s . La fonction à minimiser devient :

$$\begin{aligned} f(q) &= f(q_1, \dots, q_s) \\ &= \sum_{s=1..S} \sum_{i=1..I_s} \sum_{\{\mathcal{D}_{s,i,j}\}} d^2(C_{0s}(q_1, \dots, q_s) \cdot (P_{s,i})_{R_{seg_s}}, (\mathcal{D}_{s,i,j})_{R_{abs}}) \end{aligned}$$

où I_s est le nombre d'indices sur le segment s .

Après avoir déterminé les valeurs articulaires courantes, on peut commander le robot pour l'envoyer à la position désirée en imposant aux moteurs des vitesses

Chapitre 8. Limites et extensions possibles

proportionnelles à l'erreur commise : $\dot{q} = k(q^* - q)$. Ceci revient à proposer un contrôleur qui utilise dans sa boucle d'asservissement les informations fournies par la vision au lieu des informations fournies par les codeurs. Il devra bien sûr assurer toutes les autres fonctionnalités d'un contrôleur classique, en particulier la gestion des singularités.

Cette manière de procéder nécessite le calcul complet des valeurs articulaires courantes q ce qui risque d'être très long et donc de rendre le contrôle difficile. Aussi, plutôt que de contrôler les variables articulaires, on va essayer d'effectuer un contrôle directement sur la position 3D des indices.

8.3.3.2 Asservissement en position

Soit $p_{s,i}$ la position dans le repère R_{abs} de l'indice $P_{s,i}$ exprimé dans le repère R_{seg_s} :

$$p_{s,i} = C_{0s}(q_1, \dots, q_s) \cdot (P_{s,i})_{R_{seg_s}}$$

Soit p le vecteur constitué de la concaténation de tous les points $p_{s,i}$ dont on peut connaître la position grâce à la vision :

$$\begin{aligned} p^t &= (p_{1,1}^t, \dots, p_{1,I_1}^t, p_{2,1}^t, \dots, p_{2,I_2}^t, \dots, p_{S,1}^t, \dots, p_{S,I_S}^t) \\ &= (x_{1,1}, y_{1,1}, z_{1,1}, \dots, x_{S,I_S}, y_{S,I_S}, z_{S,I_S}) \end{aligned}$$

Soit p^* le vecteur p désiré. Alors le schéma de contrôle est celui de la figure 8.5.

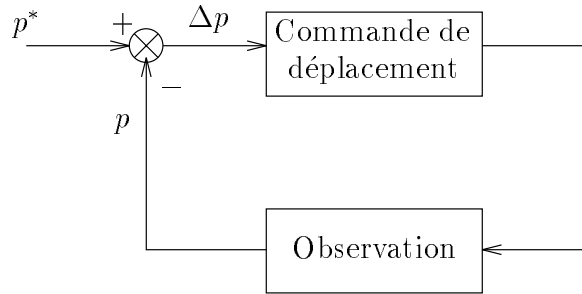


FIG. 8.5 - Asservissement en position

On veut imposer à p une vitesse proportionnelle à l'erreur commise, c'est-à-dire :

$$\dot{p} = k(p^* - p)$$

$$\text{Or : } \quad \dot{p} = \frac{\partial p}{\partial q} \dot{q} = J \dot{q} \quad \text{où } J = \begin{pmatrix} \frac{\partial x_{1,1}}{\partial q_1} & \cdots & \frac{\partial x_{1,1}}{\partial q_S} \\ \frac{\partial y_{1,1}}{\partial q_1} & \cdots & \frac{\partial y_{1,1}}{\partial q_S} \\ \frac{\partial z_{1,1}}{\partial q_1} & \cdots & \frac{\partial z_{1,1}}{\partial q_S} \\ \vdots & & \vdots \\ \frac{\partial x_{s,I_S}}{\partial q_1} & \cdots & \frac{\partial x_{s,I_S}}{\partial q_S} \\ \frac{\partial y_{s,I_S}}{\partial q_1} & \cdots & \frac{\partial y_{s,I_S}}{\partial q_S} \\ \frac{\partial z_{s,I_S}}{\partial q_1} & \cdots & \frac{\partial z_{s,I_S}}{\partial q_S} \end{pmatrix}$$

J peut être appelée “matrice d’interaction”. Elle exprime les variations des signaux, en l’occurrence les positions 3D des indices, en fonction des variations des valeurs articulaires. Elle dépend bien sûr des dérivées partielles par rapport à $q_1, q_2, \dots, q_S$ des changeurs partiels $C_{01}, C_{02}, \dots, C_{0S}$ du robot. Notons que J est une matrice triangulaire inférieure par blocs, puisque tous les termes $\frac{\partial x_{s,i}}{\partial q_j}, \frac{\partial y_{s,i}}{\partial q_j}$ et $\frac{\partial z_{s,i}}{\partial q_j}$ avec $s > j$ sont nuls.

Pour déterminer la commande $\dot{q}$ à appliquer au robot, on doit alors résoudre une équation de la forme :

$$J \cdot \dot{q} = k(p^* - p)$$

$$\text{où } \begin{cases} p & \text{est le vecteur courant dont la mesure de chaque élément a été effectuée} \\ & \text{grâce à la vision, de la même manière que pour l’asservissement articulaire} \\ & \text{(deux premières étapes),} \\ p^* & \text{est le vecteur } p \text{ désiré dont les valeurs sont déterminées grâce au calibrage} \\ & \text{des marqueurs et aux changeurs partiels du robot } C_{0s} \text{ instanciés avec les} \\ & \text{valeurs articulaires désirées } q^*. \end{cases}$$

La solution de cette équation aux moindres carrés fournit les vitesses articulaires à imposer au robot :

$$\dot{q} = k J^\dagger (p^* - p)$$

où $J^\dagger$ est la pseudo-inverse de la matrice J et est donnée par :

$$J^\dagger = (J^t J)^{-1} J^t$$

Cependant, ne connaissant pas précisément les valeurs articulaires q_i , on ne peut pas calculer explicitement la matrice J . En revanche, on peut en avoir une approximation $\hat{J}$ en l’estimant pour les valeurs q^* si elles sont suffisamment proches de q . Lors d’un suivi de trajectoire, on peut imaginer de commander le robot de proche en proche, c’est-à-dire que la trajectoire est calculée pour déplacer le robot d’une configuration initiale q_{init} à une configuration finale q_{fin} et est discrétisée en des valeurs q^* successives relativement proches les unes des autres. q_{init} peut être estimée soit de la manière décrite au paragraphe 8.3.3.1 concernant l’asservissement

Chapitre 8. Limites et extensions possibles

articulaire, soit en conservant les valeurs articulaires renvoyées par les codeurs juste avant la détection de la panne.

Dans [Esp95], on peut voir qu'une condition suffisante de décroissance du module de l'erreur au carré, $\|p^* - p\|^2$, est que la matrice $J\hat{J}^\dagger$ soit positive. Ceci sera vérifié si $\hat{J}$ est une bonne approximation de J .

L'inconvénient majeur de cette approche est qu'elle nécessite encore la reconstruction 3D des indices dans R_{abs} , ce qui peut être délicat dans le cas où ils ne sont vus que par une seule caméra. On va donc essayer de raisonner directement sur les informations contenues dans les images.

8.3.3.3 Asservissement visuel

Ici, on ne cherche plus à réguler une information 3D (la position des indices dans l'espace), mais une information 2D (la position des indices dans les images).

Soit $w_{s,i,j} = (u_{s,i,j}, v_{s,i,j})$ les coordonnées de la projection de $p_{s,i}$ dans l'image j . Alors :

$$w_{s,i,j} = M_j \cdot p_{s,i} = M_j \cdot C_{0s}(q_1, \dots, q_s) \cdot (P_{s,i})_{R_{seg_s}}$$

où M_j est la matrice de projection de la $j^{\text{ème}}$ caméra⁷. On peut choisir comme modèle de caméra un simple modèle à sténopé qui fournira une précision suffisante pour effectuer un asservissement visuel du robot.

Soit w le vecteur constitué de la concaténation de tous ces points 2D :

$$w^t = (w_{1,1,j_1}^t, \dots, w_{s,i,j}^t, \dots, w_{S,I_S,j_n}^t) = (u_{1,1,j_1}, v_{1,1,j_1}, \dots, u_{S,I_S,j_n}, v_{S,I_S,j_n})$$

Soit w^* le vecteur w désiré. Alors l'asservissement du robot se fait selon le schéma 8.6.

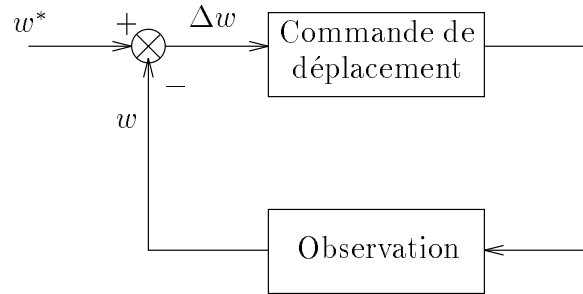


FIG. 8.6 - Asservissement visuel

On veut imposer à w une vitesse proportionnelle à l'erreur commise, c'est-à-dire :

$$\dot{w} = k(w^* - w)$$

⁷. $w_{s,i,j}$ n'existe pas forcément, puisque la caméra j ne voit pas forcément l'indice $P_{s,i}$.

$$\text{Or: } \dot{w} = \frac{\partial w}{\partial p} \frac{\partial p}{\partial q} \dot{q} + \frac{\partial w}{\partial \theta} \dot{\theta} = J_1 J_2 \dot{q} + \frac{\partial w}{\partial \theta} \dot{\theta} = J \dot{q} + \frac{\partial w}{\partial \theta} \dot{\theta}$$

où θ est le vecteur des paramètres qui caractérisent l'orientation des caméras dans le cas où elles sont orientables. La “matrice d'interaction” J exprime les variations des signaux, en l'occurrence les positions 2D des indices, en fonction des variations des valeurs articulaires. J_1 dépend bien sûr des coefficients des matrices de projection M_j . J_2 dépend des dérivées partielles des changeurs partiels $C_{01}, C_{02}, \dots, C_{0s}$ du robot, par rapport à $q_1, q_2, \dots, q_s$. Dans ce cas J est encore une matrice triangulaire inférieure par blocs.

Pour déterminer la commande $\dot{q}$ à appliquer au robot, on doit alors résoudre une équation de la forme :

$$J \dot{q} = k(w^* - w) - \frac{\partial w}{\partial \theta} \dot{\theta}$$

où $\begin{cases} w & \text{est le vecteur courant dont la mesure de chaque élément a été effectuée} \\ & \text{grâce au traitement des images,} \\ w^* & \text{est le vecteur } w \text{ désiré dont les valeurs sont déterminées grâce au calibrage} \\ & \text{des marqueurs, aux changeurs partiels du robot } C_{0s} \text{ instanciés avec les} \\ & \text{valeurs articulaires désirées } q^* \text{ et aux matrices de projection } M_j. \end{cases}$

La solution de cette équation aux moindres carrés fournit les vitesses articulaires à imposer au robot :

$$\dot{q} = J^\dagger \cdot \left(k(p^* - p) - \frac{\partial w}{\partial \theta} \dot{\theta} \right)$$

Comme dans le cas de l'asservissement en position, on est obligé d'utiliser une estimation $\hat{J}$ de la matrice J car on ne connaît pas de manière précise les valeurs articulaires q_k . Dans le cas de caméras fixes, on a $\dot{\theta} = 0$, ce qui induit une formulation similaire à celle de l'asservissement en position. Alors la convergence sera également assurée si la matrice $J \hat{J}^\dagger$ est positive ce qui est vérifié si $\hat{J}$ est une bonne approximation de J .

Les trois méthodes exposées successivement demandent de moins en moins de travail en ce qui concerne l'extraction d'informations à partir des images, mais de plus en plus en ce qui concerne la détermination du vecteur désiré. Or il est plus délicat de déterminer les valeurs articulaires ou les positions 3D des marqueurs à partir des informations visuelles, que d'estimer les valeurs visuelles à partir des valeurs articulaires désirées. L'approche par asservissement visuel est donc la plus séduisante, d'autant plus que des études [Esp95, Dor95] ont montré qu'elle ne nécessitait pas une très grande précision pour converger. Il serait intéressant d'évaluer la précision dont nous avons besoin et de mener une étude expérimentale dans le cadre qui nous est imposé.

Conclusion

Rappelons que l'objectif général de ce travail est de permettre *le guidage robotisé de gestes médico-chirurgicaux dans des conditions de sécurité maximale*. Plus généralement, il s'adresse à tout contexte d'utilisation de robots pour lequel la sécurité est prépondérante, les manipulations en milieu radioactif en étant un exemple.

Au cours de cette thèse, nous avons étudié de nombreux outils de vision et de la robotique, que nous avons intégrés dans un système complet de contrôle redondant de robots par capteurs externes. Au point de vue pratique, cet objectif se traduit par la nécessité d'effectuer des mesures précises. Aussi nous sommes-nous attachés à évaluer quantitativement les résultats de chacune des étapes impliquées.

Ceci nous a conduit à mettre en place de nombreuses expérimentations. Nous avons en particulier mené des expérimentations portant sur la totalité du projet. Elles en ont montré la faisabilité et ont permis de se faire une idée de la précision relative atteignable : 1.0 mm dans le cadre d'une application de type médical ; 3.0 mm pour une application de type nucléaire.

Nous rappelons maintenant le travail réalisé et les principaux résultats obtenus.

Nous avons répertorié en détail les méthodes de calibrage intrinsèque de caméras. Nous avons opté pour une méthode des deux plans qui ne suppose pas la concourance des droites de rétroprojection et nous avons montré expérimentalement qu'une modélisation polynômiale cubique de chaque plan était suffisante.

Le choix de cette méthode souple a permis une extension très simple aux cas où l'un des paramètres de la caméra est variable. Nous avons fait des tests pour une variation de la focale, car c'est elle qui induit les changements les plus significatifs dans l'image. Le matériel employé était un caméscope standard du commerce. La précision moyenne que nous avons obtenue pour un volume de $80 \times 70 \times 70\text{ cm}^3$ était de 0.35 mm dans le cas de paramètres fixes, et de 1.2 mm dans le cas d'une focale variable. Cette différence relativement importante se justifie en partie par des conditions d'expérimentation légèrement différentes, à la limite du champ observé par l'Optotrak qui repérait les plans de calibrage. D'ailleurs, au contraire de l'Optotrak, le système de repérage mécanique fournit des résultats très similaires dans les deux cas : respectivement 0.27 et 0.25 mm d'erreur moyenne.

Cette méthode purement mathématique, serait bien sûr aisément applicable à la variation d'autres paramètres, en particulier la mise au point ou l'ouverture. On peut même envisager la variation simultanée de plusieurs paramètres, les contraintes

Conclusion

étant la quantité importante de données de calibrage nécessaires et la complexité croissante de la modélisation de chaque plan.

Les approches reposant sur des modèles multiplans permettent un passage simple de l'image à l'espace alors que le passage de l'espace à l'image est plus complexe. Cette manière de procéder nous a donc amené à raisonner dans l'espace plutôt que dans l'image, ce qui est bien sûr équivalent, mais présente l'intérêt de fournir des mesures concrètement plus significatives : on parle en millimètres, et non pas en pixels.

Notre formulation du recalage de caméra en découle logiquement, puisque nous cherchons à minimiser une distance entre des points de l'espace et leur droite de rétroprojection respective, et non pas entre des points ou segments dans l'image, comme c'est souvent le cas dans la littérature.

Notre première étape d'initialisation ne fonctionne pas encore, vraisemblablement à cause de la sensibilité au bruit de la méthode retenue. Malgré tout, la méthode de minimisation que nous avons mise en place fournit des résultats finaux satisfaisants puisque les résidus moyens sont en général inférieurs à 1.0 mm .

Nous proposons une alternative dans le cas de l'application médicale pour calibrer les caméras directement dans le repère du robot. L'inconvénient majeur de cette méthode est qu'elle implique le robot. Aussi les erreurs du robot se répercutent-elles directement sur la qualité du calibrage des caméras. Ainsi nous avons obtenu des erreurs moyennes de 0.8 mm pour un volume de travail de $30 \times 20 \times 20\text{ cm}^3$, ce qui est moins bon que les résultats du calibrage intrinsèque effectué de manière classique.

Nous avons montré que le calibrage des marqueurs revient à résoudre un système d'équations matricielles de la forme : $A_i.X = X.B_i$. Nous avons étudié particulièrement deux méthodes classiques de résolution, que nous avons optimisées au moyen d'une minimisation d'erreur sur un ensemble important de données. Cette minimisation repose sur une fonction d'erreur originale qui permet de quantifier la ressemblance de deux matrices de passage.

Nous avons évalué les performances de cet algorithme global grâce à des données simulées bruitées. Ces tests nous ont montré que, utilisées ainsi, les deux méthodes étudiées fournissent des résultats équivalents. En effectuant un parallèle entre les valeurs de la fonction d'erreur pour des données simulées et pour des données réelles, nous avons déduit que la précision du calibrage des marqueurs est meilleure que le millimètre pour un robot Puma 260.

En ce qui concerne la détection des marqueurs dans les images, nous avons étudié et classifié les méthodes de détection subpixel. Nous avons retenu l'une d'entre elles pour déterminer précisément la position des trois croix qui constituent nos marqueurs. Notre travail dans ce domaine a été essentiellement pratique : recherche et test de matériaux permettant une bonne visibilité des marqueurs dans les images, afin d'en faciliter le traitement.

D'ailleurs le contexte applicatif de cette thèse dans son intégralité présente de

nombreuses contraintes pratiques comme nous avons pu le voir au chapitre 8. Le choix des caractéristiques des caméras et des marqueurs, ainsi que de leur disposition, est un problème à part entière qui n'a pas de solution générale. Nous y avons apporté des éléments de réflexion dans le cas des deux applications auxquelles nous nous sommes intéressés.

Le projet nécessite encore du travail afin d'intégrer les informations en provenance de toutes les caméras. Cette gestion de haut niveau aura vraisemblablement recours à une architecture spécialisée et permettra de rendre plus sûres les informations obtenues de manière indépendante par les caméras.

Nous avons également vu que l'une des extensions envisageables de ce travail consisterait à commander le robot par un asservissement visuel, ce qui permettrait de s'affranchir des informations articulaires en cas de panne, ou d'utiliser un robot dans des conditions limites ne correspondant pas aux conditions de calibrage. Cette extension est possible car elle s'appuie sur des "briques" identiques à celles que nous avons développées au cours de cette thèse. Ces mêmes briques couvrent également le domaine très large de la localisation 3D passive qui répondrait à des besoins de suivi d'objets (instruments, capteurs ...) dans des applications de gestes médico-chirurgicaux.

Enfin, l'adjonction de ce système de contrôle redondant par capteurs externes, à un robot de guidage de gestes, tel que le robot à sécurité passive, constitue un élément essentiel de sécurité pour la réalisation de gestes chirurgicaux assistés par ordinateur.

Par rapport à la solution consistant à multiplier les capteurs internes, composants divers et procédures de tests, cette approche présente les avantages suivants :

- Elle est applicable à tout système robotisé existant sans modification majeure.
- Elle ne nécessite pas la connaissance et la modélisation précise du système de guidage puisqu'elle cherche à détecter d'abord les conséquences globales des pannes - déviation par rapport à la trajectoire prévue dans l'espace - et non pas à traquer localement le comportement de chacun des éléments susceptibles de mal fonctionner.

Conclusion

Annexe A

Description du localisateur Optotrak

L'Optotrak est un localisateur 3D qui permet de suivre en temps réel les mouvements d'objets munis de diodes infrarouges. Il est composé de trois caméras CCD 1D, d'une série de diodes infrarouges, d'une unité centrale et d'un ordinateur hôte.

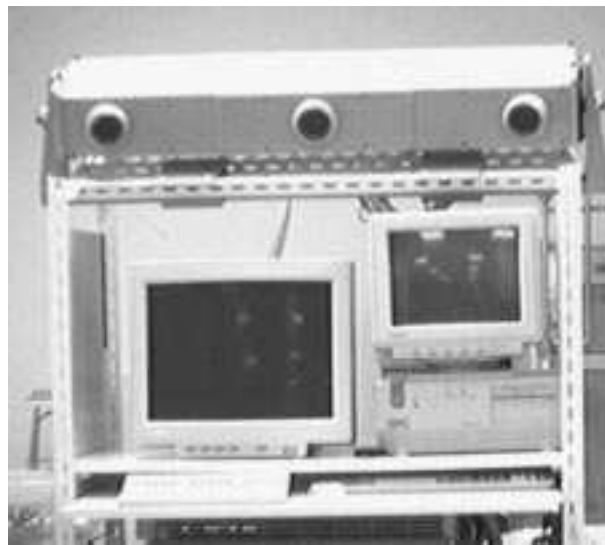


FIG. A.1 - *L'Optotrak*

Les diodes, activées l'une après l'autre sont observées par les trois caméras. L'unité centrale synchronise l'acquisition d'images avec l'activation d'une diode et reconstruit la position 3D de la diode émettrice courante. L'ordinateur est chargé de visualiser les positions en temps réel et de recevoir et stocker les données sur le disque.

Les diodes sont regroupées en "rigid bodies" (voir photo A.2).

Chaque "rigid body" est constitué d'au moins trois diodes, fixes les unes par rapport aux autres. Une procédure de calibrage permet d'associer un repère au

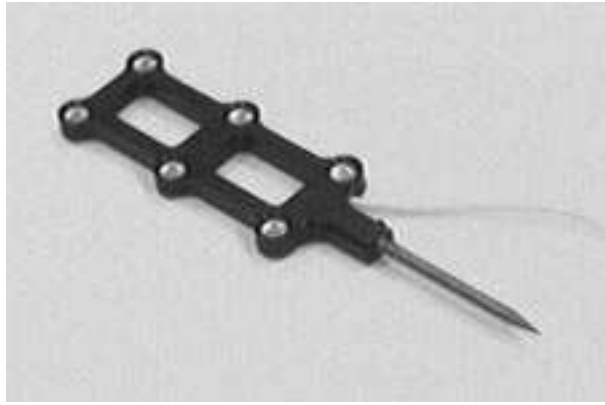


FIG. A.2 - Un “rigid body” optotrak composé de 6 diodes et muni d’une pointe

“rigid body” grâce aux positions relatives des diodes. Un “rigid body” donné peut alors être repéré par rapport à un repère absolu associé aux trois caméras, ou par rapport à un autre “rigid body”. Lorsqu’un “rigid body” est équipé de plus de trois diodes, la redondance des informations améliore la précision de sa localisation.

Un “rigid body” muni d’une pointe est appelé un palpeur. La position de la pointe dans le repère du “rigid body” peut être connue au moyen d’une procédure de calibrage qui consiste à faire pivoter le palpeur autour de sa pointe et à enregistrer un ensemble de positions dans le repère absolu. La pointe est le point exprimé dans le repère du “rigid body”, qui reste fixe dans le repère absolu. Une fois la pointe calibrée, elle peut servir à désigner des points dans l’espace.

Comme tout localisateur optique, l’Optotrak se heurte à des problèmes d’occlusion. D’autre part, il est aisément perturbé par la présence d’une source lumineuse dans le champ des caméras.

Néanmoins, utilisé dans des conditions favorables, l’Optotrak présente les caractéristiques suivantes :

- Le champ observé est d’environ $1.0 \times 1.0 \text{ m}^2$ à une distance de 2 m, $2.0 \times 2.5 \text{ m}^2$ à une distance de 4 m et de $2.5 \times 3.5 \text{ m}^2$ à une distance de 6 m.
- Nos expérimentations ont montré qu’à une distance d’environ 2.5 m, pour un champ de $0.7 \times 0.7 \text{ m}^2$, il a une répétabilité de 0.04 mm selon les axes x et y et de 0.06 mm en profondeur. Dans les mêmes conditions, sa précision relative moyenne de 0.16 mm.
- Enfin, il a une fréquence de 2500 Hz, c’est-à-dire qu’il est capable de repérer 2500 diodes par seconde. Quant à sa fréquence de localisation d’un “rigid body” composé de 6 diodes, elle peut atteindre 50 Hz.

Annexe B

Différence entre le milieu de la projection et la projection du milieu

Soit AB un segment de l'espace. On se place dans le plan contenant le segment AB et O , le centre optique de la caméra.

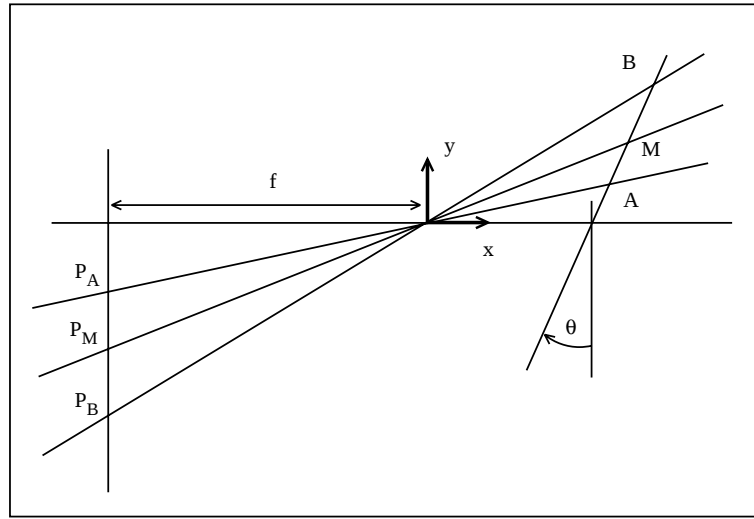


FIG. B.1 - *Modélisation pin-hole de la projection dans le plan OAB*

Soit $M(x_M, y_M)$, le milieu de $A(x_A, y_A)$ et $B(x_B, y_B)$.

Soit L , la demi-longueur du segment AB .

Soit $P_A(-f, y_{P_A})$ la projection de A .

Soit $P_B(-f, y_{P_B})$ la projection de B .

Soit $P_M(-f, y_{P_M})$ la projection de M .

Soit $M_P(-f, y_{M_P})$ le milieu de P_A et P_B .

On veut évaluer la différence entre P_M et M_P .

Annexe B. Différence entre le milieu de la projection et la projection du milieu

On exprime les coordonnées des différents points en fonction de θ, L et des coordonnées de M :

$$\begin{aligned}x_A &= x_M + L \cdot \sin \theta \\y_A &= y_M - L \cdot \cos \theta \\x_B &= x_M - L \cdot \sin \theta \\y_B &= y_M + L \cdot \cos \theta\end{aligned}$$

Le point milieu de la projection (M_P) a pour ordonnée :

$$y_{M_P} = \frac{y_{P_A} + y_{P_B}}{2} = -\frac{f \cdot y_A}{2 \cdot x_A} - \frac{f \cdot y_B}{2 \cdot x_B}$$

La projection du milieu (P_M) a pour ordonnée :

$$y_{P_M} = -\frac{f \cdot y_M}{x_M}$$

La différence entre ces 2 points est donc :

$$\begin{aligned}d &= y_{P_M} - y_{M_P} \\&= \frac{f \cdot y_A}{2x_A} + \frac{f \cdot y_B}{2x_B} - \frac{f \cdot y_M}{x_M} \\&= f \cdot \frac{y_A \cdot x_B \cdot x_M + y_B \cdot x_A \cdot x_M - 2y_M \cdot x_A \cdot x_B}{2x_A \cdot x_B \cdot x_M} \\&= \frac{f}{2} \cdot \frac{(y_M - L \cos \theta)(x_M - L \sin \theta)x_M + (y_M + L \cos \theta)(x_M + L \sin \theta)x_M - 2(x_M^2 - L^2 \sin^2 \theta)y_M}{(x_M^2 - L^2 \sin^2 \theta) \cdot x_M} \\&= f \cdot L^2 \frac{(x_M \cdot \cos \theta + y_M \cdot \sin \theta) \cdot \sin \theta}{(x_M^2 - L^2 \sin^2 \theta) \cdot x_M}\end{aligned}$$

Or la longueur du segment projeté (P_AP_B) vaut :

$$\begin{aligned}\overline{P_AP_B} &= y_{P_B} - y_{P_A} \\&= -\frac{y_B \cdot f}{x_B} + \frac{y_A \cdot f}{x_A} \\&= f \cdot \frac{y_A \cdot x_B - y_B \cdot x_A}{x_A \cdot x_B} \\&= f \cdot \frac{(y_M - L \cos \theta)(x_M - L \sin \theta) - (y_M + L \cos \theta)(x_M + L \sin \theta)}{(x_M + L \sin \theta)(x_M - L \sin \theta)} \\&= \frac{-2f \cdot L \cdot (y_M \cdot \sin \theta + x_M \cdot \cos \theta)}{x_M^2 - L^2 \sin^2 \theta}\end{aligned}$$

D'où :

$$d = \frac{L \cdot \sin \theta}{2x_M} \cdot \overline{P_AP_B} = \frac{1}{4} \cdot \frac{x_A - x_B}{x_M} \cdot \overline{P_AP_B}$$

Exemple : Soit un disque de rayon 2 cm situé à une distance $x_M = 1$ m du point focal. Dans le pire des cas $x_A - x_B$ vaut 2 cm. Alors l'erreur entre la projection du centre et le centre de la projection représente 0.5 % de la taille de la projection (si le disque projeté a une taille de 100 pixels, on a une erreur de 0.5 pixel)

Bibliographie

- [Afn93a] Afnor. Robots manipulateurs industriels - sécurité. Norme européenne - Norme française, mai 1993. NF EN 775. ISSN 0335-3931.
- [Afn93b] Afnor. Sécurité des machines - équipement électrique des machines - partie 1 : Règles générales. Norme française, mai 1993. NF EN 60204-1. ISSN 0335-3931.
- [Afn93c] Afnor. Sites de production comprenant des robots manipulateurs industriels - prévention des accidents d'origine mécanique. Norme française, mai 1993. NF E 61-120. ISSN 0335-3931.
- [AHB87] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 9(5):698–700, 1987.
- [Amo93] A. Amorison. Rapport de stage. Faculté Polytechnique de Mons, Juillet 1993.
- [ASTL94] J.R. Adler, A. Schweikard, R. Tombropoulos, and J.-C. Latombe. Image-guided robotic radiosurgery. In *MRCAS 94, Medical Robotics and Computer Assisted Surgery*, pages 291–297, Pittsburgh, USA, september 1994.
- [Aya89] N. Ayache. *Vision Stéréoscopique et Perception Multisensorielle*. InterEditions., Paris, 1989.
- [Bal81] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [BC92] R. Boie and I. Cox. An analysis of camera noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):671–674, 1992.
- [BCC95] Eric Bainville, Philippe Chaffanjon, and Philippe Cinquin. Computer generated visual assistance to a surgical operation: the retroperitoneoscopy. *Computers in Biology and Medicine*, may 1995.
- [Bey92] H.A. Beyer. Accurate calibration of ccd-cameras. In *IEEE International Conference on Computer Vision*, pages 96–101, Champaign, June 1992.

- [BFK84] W. Bohm, G. Farin, and J. Kahman. A survey of curve and surface in CAGD. *Computer Aided and Geometric Design*, 1:1–60, 1984.
- [BHIV94] G. Baardman, H.J. Hermens, M.J. Ijzerman, and P.H. Veltink. Functional Electrical Stimulation for Mobility. In *Proceedings of Robomed94, the 1st European Conference on Medical Robotics*, pages 113–119, Barcelona, Spain, June 20-22 1994.
- [BI90] Chinmoy B. Bose and Amir Israel. Design of fiducials for accurate registration using machine vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1196–1200, December 1990.
- [Bir93] G.E. Birch. Development and methodology for the formal evaluation of the neil squire foundation robotic-assistive appliance. *Robotica*, 11(6):529–534, December 1993.
- [BK87] K.L Boyer and A.C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1), 1987.
- [BKLO87] Carlos A. Berenstein, Laveen N. Kanal, David Lavine, and Eric C. Olson. A geometric approach to subpixel registration accuracy. *Computer Vision, Graphics and Image Processing*, 40:334–360, 1987.
- [BLS93] E. Bittar, S. Lavallée, and R. Szeliski. A method for registering overlapping range images of arbitrarily shaped surfaces for 3-D object reconstruction. In *SPIE Vol. 2059, Sensor Fusion VI*, Boston, September 1993. Society of Photo-Optical Instrumentation Engineers.
- [BM94] Pascal Brand and Roger Mohr. Accuracy in image measure. In *SPIE, VIDEOMETRICS III*, pages 218–228, Boston, USA, 31 October - 4 November 1994.
- [BMB94] P. Brand, R. Mohr, and Ph. Bobet. Distorsions optiques : Correction dans un modèle projectif. In *Reconnaissance des Formes et Intelligence Artificielle*, pages 87–98, Paris, Janvier 1994.
- [Car94] L. Carrat. Rapport de stage. IMERIR, Octobre 1994.
- [CDLL94] R. Cammoun, J.-M. Détriché, F. Lauture, and B. Lesigne. Robotised workstations for handicapped people. In *Proceedings of Robomed94, the 1st European Conference on Medical Robotics*, pages 131–135, Barcelona, Spain, June 20-22 1994.
- [Che91a] H.H. Chen. Pose determination from line-to-plane correspondences : Existence condition and closed-form solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):530–541, June 1991.

- [Che91b] H.H. Chen. A screw motion approach to uniqueness analysis of head-eye geometry. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 145–151, Lahaina, Maui, Hawaii, 1991.
- [CK91] J.K. Chou and M. Kamel. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *Int. J. Robotics Res.*, 10(3):240–254, June 1991.
- [CLSB92] G. Champleboux, S. Lavallée, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3-D object localization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, Champaign, Illinois, June 1992.
- [Col91] Joël Colly. *Reconnaissance d'objets articulés tridimensionnels à partir d'informations multisensorielles pour des applications robotiques*. PhD thesis, Université Paul Sabatier de Toulouse, 1991.
- [CT90] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4:127–140, 1990.
- [CVC93] A. Casals, R. Villà, and X. Cufi. Tou, an Assistant Arm : Design and Control, Objectives and Performance. In *Proceedings of '93 ICAR International Conference on Advanced Robotics*, pages 355–360, Tokyo, Japan, November 1-2 1993.
- [Dav82] Larry S. Davis. Hierarchical generalized hough transforms and line-segment based generalized hough transforms. *Pattern Recognition*, 15(4):277–285, 1982.
- [Dav93] B. Davies. Safety of medical robots. In *Proceedings of '93 ICAR International Conference on Advanced Robotics*, pages 311–317, Tokio, Japan, November 1-2 1993.
- [DF90] Rachid Deriche and Olivier Faugeras. Tracking line segments. *Image and vision computing*, 8(4):261–270, november 1990.
- [DGMDN93] P. Dario, E. Guglielmelli, C. Mulè, and M. Di Natale. URMAD : an autonomous mobile robot system for the assistance to the disabled. In *Proceedings of '93 ICAR International Conference on Advanced Robotics*, pages 341–346, Tokyo, Japan, November 1-2 1993.
- [DHN+92] B. Davies, R. Hibbert, W. Ng, A. Timoney, and J. Wickham. Mechanical constraints - the answer to safe robotic surgery? *ITBM (Innovation et Technologie en Biologie et Medecine)*, 13(4):426–436, 1992.

- [DLB91] J.M. Détriché, B. Lesigne, and T. Bernard. Development of a Workstation for Handicapped People Including the Robotized System 'MASTER'. In *International Conference on Rehabilitation Robotics*, pages 1–16, Atlanta, Georgia Institute of Technology, June 19-20 1991.
- [DMJ93] J.L. Dallaway, R.M. Mahoney, and R.D. Jackson. An interactive robot control environment for rehabilitation applications. *Robotica*, 11(6):541–551, December 1993.
- [DNH93] B. Davies, W. Ng, and R. Hibbert. Prostatic resection : an example of safe robotic surgery. *Robotica*, 11(6):561–566, December 1993.
- [Dor95] F. Dornaika. *Contribution à l'intégration Vision / Robotique : Calibrage, localisation et asservissement*. PhD thesis, Institut National Polytechnique de Grenoble, Septembre 1995.
- [DRLR89] M. Dhome, M. Richetin, J.T Lapresté, and G. Rives. Determination of the attitude of 3-d objects from a single perspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, 1989.
- [DT95] Y. Delnondedieu and J. Troccaz. Padyc : A passive arm with dynamic constraints. In *MRCAS 95, Second Annual Symposium on Medical Robotics and Computer Assisted Surgery*, pages 173–180, Baltimore, Maryland USA, November 4-7 1995.
- [Duc95] Eddy Duchêne. Localisateur video 6d pour la gmcao. Rapport de stage de l'Ecole d'Ingénieurs en Informatique Industrielle et Instrumentation, 1995.
- [Dvo83] V. N. Dvorchenko. Bounds on (deterministic) correlation functions with applications to registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):206–213, 1983.
- [EG94] Alon Efrat and Craig Gostman. Subpixel image registration using circular fiducials. *International Journal of Computational Geometry & Applications*, 4(4):403–422, 1994.
- [Esp95] B. Espiau. Sur les erreurs en asservissement visuel. Technical Report 2619, INRIA Rhône-Alpes, 46 av. Félix Viallet, 38031 Grenoble Cedex France, Juillet 1995.
- [EVF94] R. Encisco, T. Viéville, and O. Faugeras. Approximation du changement de focale et de mise au point par une transformation affine à trois paramètres. *Traitement du signal*, 11(5):361–372, 1994.

- [FB81] M.A. Fishler and R.C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [FH86] O.D. Faugeras and M. Hebert. The representation, recognition and location of 3d objects. *Int. J. Robotics Res.*, 5(3):27–52, 1986.
- [FT87] O. Faugeras and G. Toscani. Camera calibration for 3d computer vision. In *Proceedings of the International Workshop on Industrial Applications of Machine Vision and Machine Intelligence*, pages 240–247, Tokyo, Japan, February 1987.
- [Gau90] Olivier Gaudoin. *Outils statistiques pour l'évaluation de la fiabilité des logiciels*. PhD thesis, Université Joseph Fourier, Grenoble I, 1990.
- [GDLM93] R. Gelin, J.M. Détriché, J.P. Lambert, and P. Malblanc. The sprint of COACH. In *Proceedings of '93 ICAR International Conference on Advanced Robotics*, pages 325–329, Tokyo, Japan, November 1-2 1993.
- [GFE⁺95] D. Glauser, H. Fankhauser, M. Epitoux, J.-L. Hefti, and A. Jaccottet. Neurosurgical robot minerva. first results and current developments. In *MRCAS 95, Second Annual Symposium on Medical Robotics and Computer Assisted Surgery*, pages 24–30, Baltimore, Maryland USA, November 4-7 1995.
- [GQP93] A. Goswami, A. Quaid, and M. Peshkin. Identifying robot parameters using partial pose information. *IEEE Control Systems*, pages 6–14, october 1993.
- [GTK88] K. D. Gremban, C. E. Thorpe, and T. Kanade. Geometric camera calibration using systems of linear equations. In *IEEE International Conference on Robotics and Automation*, pages 562–567, Philadelphia, Pennsylvania, April 1988.
- [Gug94] E. Guglielmelli. Mobile Robots in Residential Care. In *Proceedings of Robomed94, the 1st European Conference on Medical Robotics*, pages 89–103, Barcelona, Spain, June 20-22 1994.
- [Hal79] E.L. Hall. *Computer Image Processing and Recognition*. Academic Press, New York, 1979.
- [Hav91] D. I. Havelock. The topology of locales and its effects on position uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):380–386, 1991.
- [HB93] H.J. Hermens and G. Baardman. Control of the first generation CALIES implants. In J. Edwards, editor, *Proceedings of the Conference on*

- Muscular Components in Functional Electrical Stimulation (RAFT)*, pages 52–60, 1993.
- [HD83] Peter D. Hyde and Larry S. Davis. Subpixel edge estimation. *Pattern Recognition*, 16(4):413–420, 1983.
- [HK92] G. Healey and R. Kondepudy. Ccd camera calibration and noise estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 90–95, Urbana-Champaign, Illinois, USA, June 1992.
- [HM86] A. Huertas and G. Medioni. Detection of intensity changes with sub-pixel accuracy using laplacian-gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):651–664, September 1986.
- [HM93] R. Horaud and O. Monga. *Vision par ordinateur, outils fondamentaux*. Hermes, Traité des Nouvelles Technologies, série Informatique, Paris, 1993.
- [Hol89] J.M. Hollerbach. A survey of kinematic calibration. *The Robotic Review*, MIT Press, pages 207–242, 1989.
- [Hor] Radu Horaud. Calibration caméra/pince. Rapport interne du LI-FIA (Laboratoire d’Informatique Fondamentale et d’Intelligence Artificielle), Grenoble.
- [Hor87] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A.*, 4(4):629–642, 1987.
- [Hou62] P. V. C. Hough. Methods and means for recognizing complex patterns. U.S. Patent 3069654, 1962.
- [HPH92] J. Hennequin, R. Platts, and Y. Hennequin. Putting technology to work for the disadvantaged. *Rehabilitation Robotics Newsletter*, 4(2), 1992.
- [HTR88] S. Hayati, K. Tso, and G. Roston. Improving the absolute positioning accuracy of robot manipulators. In *IEEE Robotics and Automation Conference*, pages 1400–1406, Philadelphia, January 1988.
- [Jah93] F. Jahard. Les capteurs 3d commercialisés, mémoire de fin d’études. Technical report, Telecom Paris, 1993.
- [Jar76] R. A. Jarvis. Focus optimisation criteria for computer image processing. *Microscope*, 25(2):163–180, 1976.
- [Jos94] S. H. Joseph. Unbiased least squares fitting of circular arcs. *Graphical Models and Image Processing*, 56(5):424–432, September 1994.

- [Kaw94] K. Kawamura. Prospects of research on intelligent robotic systems using flexible actuators at the intelligent robotic lab. vanderbilt university. Technical report, Intelligent Robotic Lab. Vanderbilt University, 1994.
- [KMZ⁺93] Peter Kazanzides, Brent D. Mittelstadt, Joel F. Zuhars, Phillip W. Cain, and H.A. Paul. Surgical and industrial robots : Comparison and case study. In *International Robots and Vision Automation Conference*, pages 10–19, Detroit, Michigan, 1993.
- [Kro87] E. Krotkov. Focusing. *International Journal of Computer Vision*, 1:223–237, 1987.
- [KVW94] M. Kisworo, S. Venkatesh, and G. West. Modeling edges at subpixel accuracy using the local energy approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):405–410, April 1994.
- [Lan87] U. M. Landau. Estimation of circular arc center and its radius. *Computer Vision, Graphics and Image Processing*, 38:317–326, 1987.
- [Lav89] S. Lavallée. A new system for computer assisted neurosurgery. In K. Yongmin, editor, *IEEE Engineering Medicine Biology Society (EMBS)*, pages 926–927, Seattle, November 1989.
- [LCDB91] S. Lavallée, P. Cinquin, J. Demongeot, and A.L. Benabid. Neurochirurgie stéréotaxique assistée par ordinateur et robot. *ITBM (Innovation et Technologie en Biologie et Medecine)*, 12(2):166–178, 1991.
- [Lei91] L. Leifer. Development of a Visual Language for Assistive Robot Command-Control. In *Draft of Proceedings of Rehabilitation Robotics Workshop*, University of Cambridge, Cambridge, England, April 12-14 1991.
- [LHF90] Y. Liu, T.S. Huang, and O.D. Faugeras. Determination of camera location from 2-d to 3-d line and point correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, January 1990.
- [LMAR89] Edward P. Lyvers, Owen Robert Mitchell, Mark L. Akey, and Anthony P. Reeves. Subpixel measurement using a moment-based edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1293–1309, December 1989.
- [Low91] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

- [Low92] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992.
- [LRD92] J.M. Lavest, G. Rives, and M. Dhome. Utilisation d'un objectif à focale variable en vision monoculaire en vue de la reconstruction 3d. *Traitement du signal*, 9(6):491–506, 1992.
- [LRD94] J.M. Lavest, G. Rives, and M. Dhome. Calibrage et reconstruction par invariant projectif. In Afcet, editor, *neuvième congrès Reconnaissance des Formes et Intelligence Artificielle*, pages 623–628, Paris, Janvier 1994.
- [LSB91] S. Lavallée, R. Szeliski, and L. Brunie. Matching 3-D smooth surfaces with their 2-D projections using 3-D distance maps. In *SPIE Vol. 1570 Geometric Methods in Computer Vision*, pages 322–336, San Diego, CA, July 1991.
- [LTS⁺96] S. Lavalée, J. Troccaz, P. Sautot, B. Mazier, P. Cinquin, P. Merloz, and J.P. Chirossel. Computer assisted spine surgery using anatomy-based registration. In R. Taylor, S. Lavalée, G. Burdea, and R. Mosges, editors, *Computer Integrated Surgery*, chapter 32. MIT Press, Cambridge, MA, Cambridge, 1996.
- [MBK81] H. A. Martins, J. R. Birk, and R. B. Kelley. Camera models based on data from two calibration planes. *Computer Graphics and Image Processing*, 17:173–179, 1981.
- [MH95] N.J. Maitland and C.G. Harris. A video based tracker for use in computer aided surgery. In BMVA Press E.R. Hancock, editor, *Proceedings of British Machine Vision Conference*, pages 609–618, September 1995.
- [MKM⁺95] Ken Masamune, Etsuko Kobayashi, Yoshitaka Masutani, Makoto Suzuki, Takeyoshi Dohi, Hiroshi Iseki, and Kintomo Takahura. Development of a mri compatible needle insertion manipulator for stereotactic neurosurgery. In *MRCAS 95, Second Annual Symposium on Medical Robotics and Computer Assisted Surgery*, pages 165–172, Baltimore, Maryland USA, November 4-7 1995.
- [Mor95] H. Mortimore. Reconstruction euclidienne d'une scène tridimensionnelle avec une caméra affine non calibrée montée sur un bras de robot. Rapport de DEA d'informatique de l'IMAG, Juin 1995.
- [MR84] J.P. Micaleff and P. Rabishong. Motorized assistance for the handicapped: control systems, vehicules, walking machine. In A. Pedotti and R. Andrich, editors, *Proceedings of the First European Workshop on*

- Evaluation of Assistive Devices for Paralyzed Persons*, pages 125–132, COMAC BME, 1984.
- [MRRO94] Dennis A. Montera, Steven K. Rogers, Dennis W. Ruck, and Mark E. Oxley. Object tracking through adaptative correlation. *Optical Engineering*, 33(1):294–302, January 1994.
- [NN90] S.K. Nayar and Y. Nakagawa. Shape from focus : An effective approach for rough surface. In *IEEE International Conference on Robotics and Automation*, pages 218–225, 1990.
- [NS89] Eric Niel and Jean-Pierre Simon. *La sécurité d’exploitation des installations robotisées*, volume 20 of *Technologies de pointe*. Hermes, Paris, 1989.
- [OLJ⁺93] R. Orti, S. Lavallee, R. Julliard, P. Cinquin, and E. Carpentier. Computer assisted knee ligament reconstruction. In *IEEE Engineering in Medicine and Biology Society Proceedings*, pages 936–937, San Diego, 1993.
- [Pa90] I. Perlash and al. Clinical evaluation of a vocational desktop robotic aid for severely physically disabled individuals. Technical report, Rehabilitation Research and Development, Department of Veterans Affairs, 1990. est-ce bien un rapport technique?
- [Per91] José-Philippe Perez. *Optique géométrique, ondulatoire et polarisation*. Masson, Paris, 1991.
- [Peu93] B. Peuchot. Camera virtual equivalent model 0.01 pixel detectors. *Computerized Medical Imaging and Graphics*, 17:289–294, 1993.
- [PFTV92] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, second edition, 1992.
- [Pic80] B. Picinbono. *Éléments de théorie du signal*. Dunod, Paris, 1980.
- [PKD83] J. A. Parker, R. V. Kenyon, and D. E. Droxel. Comparison of interpolating methods for image resampling. *IEEE Transactions on Medical Imaging*, 2(1):31–39, 1983.
- [Poy94] A. Poyet. Contrôle redondant de la position d’un robot : Etude et proposition d’une solution. Rapport interne TIMC-EDF, Septembre 1994.
- [PSMP91] C.J. Pournaras, R.D. Shonat, J.L. Munoz, and B.L. Pretig. New ocular micromanipulator for measurements of retinal and vitreous physiologic parameters in the mammalian eye. *Experimental Eye Research*, 53:723–727, 1991.

- [PTP87] Jocelyne Pertin-Troccaz and Pierre Puget. Dealing with uncertainty in robot planning using proving techniques. In MIT Press, editor, *4th International Symposium of Robotic Research*, Santa Cruz, 1987.
- [PVG93] J.P. Prenninger, M. Vincze, and H. Gander. Contactless position and orientation measurement of robot end-effectors. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 180–185, 1993.
- [Ra91a] Taylor R.H. and al. A model-based optimal planning and execution system with active sensing and passive manipulation for augmentation of human-precision in computer integrated surgery. In *Proc. of the Int. Symposium on Experimental Robotics*, 1991.
- [Ra91b] J.C. Rosier and all. Rehabilitation Robotics : The MANUS Concept. In '91 ICAR, *Proceedings of the Fifth International Conference on Advanced Robotics, Robots in Unstructured Environments*, pages 893–898, Pisa, Italy, June 19-22 1991.
- [RA94] V. Rodin and A. Ayache. Stéréovision axiale : modélisation et calibrage du système de prises de vues, reconstruction 3d d'objets naturels. *Traitement du signal*, 11(5):373–391, 1994.
- [RDW93] Bobby S. Rao and Hugh Durrant-Whyte. A decentralized bayesian algorithm for identification of tracked targets. *IEEE Transactions on Systems, Man and Cybernetics*, 23(6):1683–1698, November/December 1993.
- [RSR94] K. Radermacher, H.W. Staudte, and G. Rau. Computer assisted orthopaedic surgery by means of individual templates - aspects and analysis of potential applications. In *MRCAS 94, Medical Robotics and Computer Assisted Surgery*, pages 42–48, Pittsburgh, USA, september 1994.
- [SA87] Y.C. Shiu and S. Ahmad. Finding the position of a sensor by solving a homogeneous equation of the form $a.x=x.b$. In *IEEE International Conference on Robotics and Automation*, pages 1666–1671, Raleigh, 1987.
- [Sau94] P. Sautot. *Vissage Pédiculaire Assisté par Ordinateur*. PhD thesis, Université Joseph Fourier, September 1994.
- [SH92] C.A. Shaffer and G.M. Herb. A real-time robot arm collision avoidance system. *IEEE Transactions on Robotics and Automation*, 8(2):149–160, 1992.

- [SYI86] K. Sato, H. Yamamoto, and S. Inokuchi. Tuned range finder for high precision 3d data. In *8th International Conference on Pattern Recognition*, Paris, 1986.
- [TC89] Samuel M. Thomas and Y.T. Chan. A simple approach for the estimation of circular arc center and its radius. *Computer Vision, Graphics and Image Processing*, 45:362–370, 1989.
- [TC94] M. Tichem and M. S. Cohen. Sub μ m registration of fiducial marks using machine vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):791–794, 1994.
- [TCK⁺91] R.H. Taylor, C.B. Cutting, Y. Kim, A. Kalvin, D. Larose, B. Haddad, D. Khoramabadi, M. Noz, R. Olyha, N. Bruun, and D. Grimm. A Model-Based Optimal Planning and Execution System with Active Sensing and Passive Manipulation for Augmentation of Human Precision in Computer-Integrated Surgery. In *Proceedings of the Second International Symposium on Experimental Robotics*, LAAS/CNRS, Toulouse, France, June 25-27 1991.
- [TFLT92] R.H. Taylor, J. Funda, D. LaRose, and M. Treat. A telerobotic system for augmentation of endoscopic surgery. In *IEEE Engineering in Medicine and Biology Society*, pages 1054–1056, 1992.
- [TG94] J.Ph. Tarel and A. Gagalowicz. Calibration de caméra à base d'ellipses. In *Reconnaissance des Formes et Intelligence Artificielle*, pages 209–220, Paris, Janvier 1994.
- [TH86] Qi Tian and Mickael N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics and Image Processing*, 35:220–233, 1986.
- [TKMP90] R.H. Taylor, P. Kazanzides, B.D. Mittelstadt, and H.A. Paul. Redundant Consistency Checking in a Precise Surgical Robot. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 12, No. 5*, pages 1933–35, 1990.
- [TL88] R. Y. Tsai and R.K. Lenz. Real time versatile hand/eye calibration using 3D machine vision. In *IEEE Conference on Robotics and Automation*, pages 554–561, Philadelphia, 1988.
- [TM84] Ali T. Tabatabai and Owen Robert Mitchell. Edge location to subpixel values in digital imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):188–201, March 1984.
- [Top93] M. Topping. Early experience in the use of the 'handy 1' robotic aid to eating. *Robotica*, 11(6):525–527, December 1993.

- [TPC⁺92] R. Taylor, H. Paul, C. Cutting, B. Mittelstadt, W. Hanson, P. Kazantzides, B. Musits, Y. Kim, A. Kalvin, B. Haddad, D. Khoramabadi, and D. Larose. Augmentation of human precision in computer-integrated surgery. *ITBM (Innovation et Technologie en Biologie et Medecine)*, 13(4):450–468, 1992.
- [Tsa87] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.
- [TTG92] K. Tarabanis, R.Y. Tsai, and D.S. Goodman. Modeling of a computer-controlled zoom lens. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1545–51, Nice, France, May 1992.
- [ULWW94] D.R. Uecker, C. Lee, Y.F. Wang, and Y. Wang. A speech-directed multi-modal man-machine interface for robotically enhanced surgery. In *MRCAS 94, First Annual Symposium on Medical Robotics and Computer Assisted Surgery*, pages 176–183, Pittsburgh, USA, September 1994.
- [Vai90] R. Vaillant. *Géométrie différentielle et vision par ordinateur : détection et reconstruction des contours d'occultation de la surface d'un objet non-polyédrique*. PhD thesis, Université Paris-Sud, 1990.
- [VdLML86] H.F. Van der Loos, S.J. Michalowski, and L.J. Leifer. Design of an omnidirectional mobile robot as a manipulating device for the severely disabled. In *Proceedings of the 9th Annual Conference on Rehabilitation Engineering (RESNA)*, Minneapolis, MN, 1986.
- [VO90] P. Vuylsteke and A. Oosterlinck. Range image acquisition with a single binary-encoded light pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2), 1990.
- [Wag95] Guido Wagener. Le système crigos. Rapport de stage, 1995.
- [Wan92] Ching-Cheng Wang. Extrinsic calibration of a vision sensor mounted on a robot. *IEEE Transactions on Robotics and Automation*, 8(2):161–175, April 1992.
- [Wen71] Magnus J. Wenninger. *Polyhedron Models*. Cambridge University Press, Cambridge, 1971.
- [Wil94] R.G. Willson. Modeling and calibration of automated zoom lenses. In *Proceedings of the SPIE vol. 2350: Videometrics III*, Boston, MA, October 1994.

- [Woh92] T.T. Wohlers. 3d digitizers. *Computer Graphics World*, pages 73–77, 1992.
- [WW90] A.G. Wiley and K.W. Wong. Metric aspects of zoom vision. In *Proceedings of the SPIE vol. 1395: Close-Range Photogrammetry Meets Machine Vision*, pages 112–118, Zurich, Suisse, 1990.
- [Yua89] J.S.C. Yuan. A general photogrammetric method for determining object position and orientation. *IEEE Transactions on Robotics and Automation*, 5(2):129–142, 1989.
- [ZS93] Hanqi Zhuang and Yiu Cheung Shiu. A noise-tolerant algorithm for robotic hand-eye calibration with or without sensor orientation measurement. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):1168–1175, July/August 1993.
- [ZSM93] S. Zhao, A. Schmitt, and D. Meizel. Ultrasonic simulation in a large scope of incident angles and visible environment dynamical modelling for a mobile robot using a «multirings» configuration. In *'93 International Conference on Advanced Robotics*, pages 573–578, November 1-2 1993.