No d'Ordre : 1393

THESE

Présentée devant

L'Université de Rennes I

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

Pour obtenir

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

$\begin{array}{c} & par \\ Farabi \ BENSALAH \end{array}$

Titre de la thèse

Estimation du mouvement par vision active

Soutenue le 11 Juillet 1996 devant la commission d'examen

MM.	Jean-Jacques	Fuchs	Président
	Bernard	Espiau	Rapporteur
	Thierry	Viéville	Rapporteur
	Gabriel	Abba	Examinateur
	Patrick	Bouthemy	Examinateur
	François	Chaumette	Examinateur
	Jacques	Labiche	Examinateur

A mes Parents

Remerciements

Je tiens à remercier Monsieur Jean-Jacques Fuchs, Professeur à l'Université de Rennes 1, qui m'a fait l'honneur de présider mon jury de thèse.

Mes remerciements vont aussi à :

Messieurs Bernard Espiau, Directeur de Recherche à l'I.N.R.I.A. Grenoble et Thierry Viéville, Chargé de Recherche à l'I.N.R.I.A. Sophia Antipolis, qui ont accepté d'être les rapporteurs de ce travail,

Monsieur Gabriel Abba, Maître de Conférences à l'Université de Strasbourg,

Monsieur Jacques Labiche, Professeur à l'Université de Caen

pour avoir bien voulu participer à mon jury de thèse et juger ma contribution au domaine de la vision robotique.

Mes sincères remerciements à :

Monsieur Patrick Bouthemy, Directeur de Recherche à l'I.N.R.I.A. Rennes pour avoir accepté d'être mon Directeur de thèse,

Monsieur François Chaumette, Chargé de Recherche à l'I.N.R.I.A. Rennes pour l'aide qu'il m'a apportée tout au long de ces années, et

Monsieur Claude Labit, Directeur de Recherche à l'I.N.R.I.A. Rennes pour m'avoir accueilli dans son équipe.

	\mathbf{Intr}	oduct	ion	5				
1	Etat de l'art en asservissement visuel							
	1.1	Tâche	s de positionnement	10				
	1.2		s de suivi	13				
	1.3		roche fonction de tâche					
2	Sch	éma d	e commande	17				
	2.1	2.1 La matrice d'interaction						
	2.2	L'app	approche fonction de tâche					
	2.3	La loi	de commande référencée vision	20				
	2.4		imentations réalisées					
	2.5	_	ge du gain proportionnel λ					
		2.5.1						
		2.5.2	Gain adaptatif	32				
			2.5.2.1 Résultats expérimentaux	35				
3	Esti	matio	n robuste aux changements de modèle du mouvement de					
ľo	bjet			43				
	3.1	Estim	ation du mouvement de l'objet dans l'image	43				
	3.2	Filtra	ge de la mesure du mouvement de la cible	47				
		3.2.1	Cas général	47				
		3.2.2	Filtre de Kalman à vitesse constante, saut en vitesse	53				
		3.2.3	Filtre de Kalman à accélération constante, saut en accélération	57				
		3.2.4	Filtre de Kalman à accélération constante, saut en vitesse ou					
			en accélération	60				
	3.3	Résult	ats expérimentaux	61				
		3.3.1	Suivi d'un carré par liaison rigide	62				
			3.3.1.1 Résultats sans estimation du mouvement de l'objet	62				

			3.3.1.2	Filtre de Kalman à vitesse constante et bruit corrélé	64
			3.3.1.3	Filtre de Kalman à accélération constante et bruit	
				corrélé	64
			3.3.1.4	Test du GLR sur un filtre de Kalman à vitesse	
				constante et bruit corrélé	66
		3.3.2	Suivi d'	un point en contrôlant l'orientation de la caméra	68
			3.3.2.1	Résultats sans estimation du mouvement de l'objet .	68
			3.3.2.2	Filtre de Kalman à vitesse constante et bruit corrélé	68
			3.3.2.3	Filtre de Kalman à accélération constante et bruit corrélé	71
			3.3.2.4	Test du GLR et filtre de Kalman à vitesse constante et bruit corrélé	71
		3.3.3	Suivi d'	un point animé d'un mouvement accéléré	74
			3.3.3.1	Résultats sans estimation du mouvement de l'objet.	74
			3.3.3.2	Filtre de Kalman à vitesse constante et bruit corrélé	74
			3.3.3.3	Filtre de Kalman à accélération constante et bruit	
				corrélé	74
			3.3.3.4	Détection de sauts en accélération avec le test	
				du GLR sur un filtre de Kalman à accélération	
				constante et bruit corrélé	78
		3.3.4	Suivi d'	un point animé d'un mouvement complexe	78
	3.4	Concl	usion		81
	T	. , .			0 =
4				ouvement 3D d'un objet par vision active	87 87
	4.1				
	4.2	/ 11			
	4.3			mouvement 3D d'un point	95
		4.3.1		uvaise piste!	98
		4.3.2		es cas particuliers	99
			4.3.2.1	Cas d'un objet immobile	99
			4.3.2.2	Cas où les mouvements de la caméra et de l'objet	00
			4202	sont coplanaires	99
			4.3.2.3	De l'impossibilité à remonter au mouvement 3D de	
				l'objet dans le cas d'un changement d'orientation seulement	99
			4.3.2.4	De l'impossibilité à remonter au mouvement 3D de	99
			4.3.2.4	±	
				l'objet dans le cas d'une caméra animée d'un seul mouvement de translation uniforme	99
	4.4	Rága la	ata da ai	mulation	
	44	-11.081111	aus ue sii	411U14GUUII	101

		4.4.1		ion à partir de deux positions fixes de la caméra .	
			4.4.1.1	Cas général, sans bruit	
			4.4.1.2	Profondeur constante, sans bruit	
			4.4.1.3	Cas général, avec bruit	103
			4.4.1.4	Profondeur constante, avec bruit	105
			4.4.1.5	Faible déplacement entre les deux positions de la ca	L-
				méra, avec bruit	106
			4.4.1.6	L'objet est à l'arrêt, sans bruit	107
			4.4.1.7	L'objet est à l'arrêt, avec bruit	108
		4.4.2	Estimat	ion à partir de deux mouvements différents de la cam	iéra109
			4.4.2.1	Cas général, sans bruit	109
			4.4.2.2	Profondeur constante, sans bruit	110
			4.4.2.3	Cas général, avec bruit	111
			4.4.2.4	Profondeur constante, avec bruit	112
			4.4.2.5	L'objet est à l'arrêt, avec bruit	114
		4.4.3	Estimat	ion à partir d'un seul mouvement de la caméra	114
			4.4.3.1	Cas général, sans bruit	115
			4.4.3.2	Cas général, vitesse dans l'image calculée sur 1 ité	<u>-</u>
				ration, sans bruit	115
			4.4.3.3	Cas général, vitesse dans l'image calculée sur 1 ité	<u>-</u>
				ration, avec bruit	117
			4.4.3.4	Cas général, avec bruit	118
			4.4.3.5	Cas général, faible déplacement entre O_1 et O_2 , ave	c
				bruit	119
			4.4.3.6	Profondeur constante, sans bruit	120
			4.4.3.7	Profondeur constante, point immobile au centre d	e
				l'image, avec bruit	122
			4.4.3.8	Quelle position du point dans l'image pour un	e
				bonne estimation de son mouvement?	123
	4.5	Résult	tats expé	rimentaux	125
		4.5.1		ion à partir de deux positions fixes de la caméra .	
		4.5.2	Estimat	ion à partir de deux mouvements différents de la cam	iéra125
		4.5.3	Estimat	sion à partir d'un seul mouvement de la caméra	127
	4.6	Concl	usion		129
С	onclu	ısion			131
A	Initi	alisati	on des f	iltres de Kalman	133
В	ibliog	graphi	e		137

Introduction

Les développements réalisés durant la dernière décennie dans le domaine de la technologie des capteurs de vision et du traitement d'image ont permis l'utilisation effective des données visuelles dans la boucle de commande des robots. Ceci a pour conséquence l'amélioration de l'exécution des tâches robotiques ou de perception. Concernant les aspects liés à la robotique, il est en effet possible de tenir compte des incertitudes et/ou des changements dans l'environnement du robot (par exemple, pour compenser des erreurs de modélisation et de calibration dans les tâches de positionnement, ou pour suivre et saisir des objets convoyés sur un tapis roulant). Concernant les aspects liés à la vision par ordinateur, il est maintenant possible, par des approches de vision active, de contrôler les paramètres d'une caméra (dont notamment le mouvement), dans le but d'améliorer la réalisation de diverses tâches de perception telles que la reconnaissance, la localisation ou la reconstruction 3D d'objets.

Les différents travaux en asservissement visuel s'intéressent soit à des objets statiques, soit à des objets mobiles. Concernant les objets statiques, la tâche à réaliser est une tâche de positionnement, généralement pour des applications d'inspection, d'assemblage ou de préhension. Par contre, pour des objets en mouvement, il s'agit d'assurer une tâche de poursuite. La réalisation de telles tâches est alors plus délicate à effectuer puisqu'elle nécessite d'estimer le mouvement de ces objets, afin d'assurer un suivi correct (c'est-à-dire sans erreur de traînage). Les applications des tâches de poursuite concernent bien évidemment le domaine de la télésurveillance, mais également la robotique manufacturière pour des tâches de montage automatique au défilé, supprimant ainsi la nécessité d'arrêter les chaînes de montage pour la fixation de pièces supplémentaires.

Le travail présenté dans cette thèse s'intéresse à l'estimation du mouvement 2D et 3D d'un objet appartenant à la scène. Il y est notamment démontré l'intérêt de l'application des techniques de détection de rupture de modèle au domaine de la poursuite de cible par asservissement visuel. Plus précisément, dans le cas de l'estimation du mouvement 2D, nous décrivons un système de poursuite basé sur le filtrage de Kalman et l'algorithme du maximum de vraisemblance généralisé (GLR) capable de détecter, d'estimer et de compenser de brusques changements de direction ou d'amplitude dans le mouvement de l'objet poursuivi. Dans notre cas particulier, nous chercherons à détecter des sauts en vitesse ou en accélération.

D'autre part, concernant le mouvement 3D d'un objet, nous avons développé une méthode permettant son estimation à partir de la mesure de sa position et de sa vitesse dans l'image, ainsi que de la mesure du mouvement de la caméra. L'estimation du mouvement 3D est un problème très classique en vision par ordinateur. Ce qui caractérise nos travaux est l'approche vision dynamique active adoptée. Comme nous le verrons, l'aspect dynamique de cette approche permet de passer outre le problème classique du facteur d'échelle. En effet, dans le cas d'un point, nous obtenons une estimation des trois paramètres du mouvement ainsi que de sa profondeur. Dans une deuxième étape, nous nous sommes intéressés à l'optimisation de la trajectoire de la caméra. Cet aspect actif permet d'obtenir une estimation non biaisée et d'améliorer l'estimation du mouvement 3D de l'objet.

La présentation de notre travail respecte le plan suivant :

- Nous commençons par faire un état de l'art des travaux récents en asservissement visuel. L'application du concept de fonction de tâche à la commande référencée vision y est notamment détaillée. Cette approche, développée à l'IN-RIA et sur laquelle est basé notre travail, permet de formaliser le problème d'exécution de tâches utilisant des données visuelles sous forme d'un problème d'automatique et de régulation dans l'image.
- Dans le second chapitre, nous présentons un schéma de commande complet qui permet explicitement de poursuivre un objet mobile. Nous y détaillons les moyens de détermination du gain proportionnel adaptatif appliqué dans notre commande. Ces calculs sont nécessaires à la bonne réalisation de la tâche.
- Au troisième chapitre, nous décrivons le schéma d'estimation du mouvement 2D de l'objet poursuivi. Dans le but de minimiser les erreurs de poursuite, nous utilisons en parallèle à un filtre de Kalman, le test du maximum de vraisemblance généralisé, un algorithme capable de détecter, d'estimer et de

compenser des variations brusques du mouvement de la cible. Les différents modèles qui peuvent être retenus sont étudiés.

• Dans le dernier chapitre, nous décrivons l'état de l'art des algorithmes d'estimation du mouvement 3D à partir de données visuelles 2D. Nous présentons ensuite les résultats que nous avons obtenus par une approche de vision active et sous l'hypothèse que la vitesse de l'objet auquel on s'intéresse reste constante (en module et direction) pendant l'estimation. Nous verrons qu'on peut déterminer un mouvement optimal à faire effectuer à la caméra en vue d'améliorer l'estimation.

Les algorithmes développés ont été validés sur la cellule de vision dynamique active de l'IRISA. Nous présentons les résultats expérimentaux correspondants à chaque fin de chapitre.

Chapitre 1

Etat de l'art en asservissement visuel

Il est souvent essentiel d'associer un capteur extéroceptif à un robot pour améliorer ses possibilités et ses performances dans l'exécution de différentes tâches. Parmi les différents capteurs possibles, la vision occupe une place privilégiée, car elle permet d'avoir des informations de grande richesse. Beaucoup de chercheurs s'y sont consacrés depuis maintenant une quinzaine d'années.

L'asservissement visuel est un domaine de recherche dont le but est d'intégrer en temps réel les informations visuelles dans la boucle d'asservissement des robots manipulateurs ou mobiles. En utilisant une caméra embarquée, les techniques d'asservissement visuel permettent de fournir une interaction directe entre la position de l'effecteur du robot et son environnement. Il en résulte une commande plus précise et plus robuste en présence d'incertitudes et d'erreurs de modélisation. La tâche de base en asservissement visuel consiste à contrôler de façon stable et robuste l'attitude 6D (position et orientation) de l'effecteur du robot, potentiellement variable dans le temps, à l'aide d'informations visuelles. De nombreuses autres tâches peuvent également être réalisées comme par exemple le seul contrôle de l'orientation de la caméra. Dans tous les cas, elles reposent sur la mesure et l'utilisation des informations visuelles extraites en temps réel des images acquises par le capteur.

L'asservissement visuel touche à un certain nombre de domaines de recherche, incluant la cinématique, la dynamique et la commande des robots, le traitement d'image en temps réel, la vision active, le calcul temps réel et l'architecture système. En tant que tel, et comme le note J. Funda dans [Funda95], c'est une discipline jeune et pleine de défis avec un grand nombre de résultats préliminaires fort intéressants,

aussi bien qu'une multitude de problèmes encore ouverts.

Dans ce chapitre, nous présentons les travaux déja effectués dans le domaine de l'asservissement visuel. L'intérêt des chercheurs s'est d'abord porté sur des tâches de positionnement avant de considérer le cas plus difficile de suivi d'objets. Nous décrirons les différents travaux qui ont été menés dans les deux cas puis nous détaillerons l'approche fonction de tâche que nous avons adoptée.

1.1 Tâches de positionnement

La première contribution importante au domaine de la commande référencée vision est dûe à Weiss et Sanderson [Weiss et al.87]. Ils ont proposé plusieurs schémas de commande permettant de réaliser des tâches de positionnement. D'une part, la stratégie dite "Look and Move" (statique ou dynamique) où les aspects de vision et de commande sont considérés comme deux problèmes disjoints, et d'autre part, les techniques d'asservissement visuel, basées sur la régulation d'informations visuelles dans l'image (par exemple, les coordonnées de plusieurs points dans l'image).

Parmi ces deux différentes approches intégrant des informations visuelles en commande (asservissement purement visuel et asservissement en attitude), la première s'avère plus intéressante. Cette approche permet en effet de s'affranchir entièrement de la phase d'interprétation des informations image (notées s), nécessaire dans l'asservissement en attitude pour estimer la position et l'orientation de la caméra par rapport à la scène (rappelons que l'attitude d'un repère par rapport à un autre, peut-être représenté par un vecteur de dimension 6, noté <u>r</u>, rassemblant les trois composantes de translation et les trois composantes de rotation du changement de repère). Les techniques d'asservissement visuel, véritable boucle fermée sur les informations 2D, consistent à spécifier le problème de l'asservissement en termes de régulation dans l'image, et ne nécessitent pas la reconstruction tridimensionnelle de la scène à chaque itération de la boucle de commande. Les consignes ne sont plus exprimées sous la forme d'une attitude désirée entre la caméra et la scène, mais sous la forme d'un motif à atteindre dans l'image : les informations visuelles \underline{s} choisies pour constituer ce motif doivent atteindre les valeurs \underline{s}^* qui correspondent à une bonne réalisation de la tâche (voir Figure 1.1). Il en résulte par conséquent un gain non négligeable en précision de positionnement et en robustesse par rapport aux erreurs de modélisation et de calibration [Espiau95].

Il est montré dans [Chaumette90] que les informations visuelles utilisables en

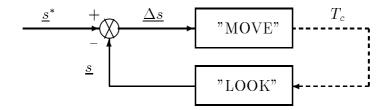


Figure 1.1 : Schéma de l'asservissement visuel

commande référencée vision peuvent être modélisées comme un ensemble \underline{s} de signaux visuels élémentaires s_j , qui dépendent uniquement de l'attitude entre la caméra et la scène. Par exemple, \underline{s} peut être constitué des coordonnées de plusieurs points, ou des paramètres décrivants des droites, des ellipses, etc . . . En considérant l'image désirée et l'image observée par la caméra, le problème de la commande se réduit à la régulation de $(\underline{s} - \underline{s}^*)$ dans l'image.

Pour utiliser les techniques d'asservissement visuel, il s'avère indispensable de modéliser le lien entre les variations dans l'image des informations visuelles et le mouvement relatif entre la caméra et la scène. Ce lien est décrit par une matrice jacobienne, également appelée matrice d'interaction. Connaissant ce lien, il est possible, d'une part, de choisir les informations visuelles capables de représenter une tâche, et d'autre part, d'élaborer des lois de commande à partir de ces informations. La commande calculée est généralement une consigne de vitesse 3D en translation, et en rotation de la caméra notée T_c (voir Figure 1.1). Son application fait bien évidemment intervenir le jacobien du robot ainsi que la relation géométrique entre la caméra et l'extrémité du robot sur laquelle elle est montée. De par son approche en boucle fermée, les tâches d'asservissement visuel sont cependant robustes aux éventuelles erreurs introduites par ces termes.

Pour les tâches de positionnement par rapport à des objets fixes, Feddema et al. ont développé une méthode pour sélectionner automatiquement les informations visuelles à prendre en compte dans la commande [Feddema89], [Feddema et al.91] dans le cas d'objets aux caractéristiques connues. Cette sélection est effectuée en fonction de l'éventuelle facilité d'extraction dans l'image des points 2D caractéristiques de l'objet, et d'un critère sur la commande (basé sur le conditionnement de la matrice jacobienne des informations visuelles). Feddema et al. ont également proposé une méthode permettant de réduire le temps de calcul du traitement d'image (et donc de la boucle d'asservissement) basée sur l'utilisation de la prédiction de la position de l'objet dans l'image [Feddema et al.90]. Ils emploient un générateur de trajectoire

des informations visuelles afin de garder ces dernières dans le champ de vue de la caméra et d'assurer une convergence lisse vers leur position désirée dans l'image.

Nelson et al. ont étudié dans [Nelson et al.93b] l'utilisation de l'asservissement visuel pour des tâches d'assemblage. Le principal centre d'intérêt de ce travail concerne la modélisation et la commande d'un système caméra-robot, la détermination automatique de l'emplacement optimal de la caméra, et le calcul adaptatif et en ligne de la transformation caméra-objet. Des travaux similaires sont présentés dans [Sharma et al.95].

Hashimoto et al. dans [Hashimoto et al.93] proposent une formulation de la commande par asservissement visuel, où le modèle est exprimé comme un système d'équations différentielles non linéaires dans l'espace d'état. Deux stratégies de commande différentes sont étudiées : d'une part, une commande linéaire obtenue par linéarisation autour du point de fonctionnement, et d'autre part, une commande non-linéaire obtenue par compensation et découplage de la dynamique du robot. De bons résultats sont obtenus en utilisant un jacobien étendu des points 2D considérés dans la commande. Le nombre de points a été choisi supérieur à trois (nombre nécessaire minimum) pour réduire la sensibilité du jacobien aux bruits de mesure. La commande linéaire a été testée sur un ensemble de tâches de positionnement à l'aide d'un Puma 560, tandis que la commande non-linéaire n'a été testée que sur un bras manipulateur à deux articulations.

Le livre intitulé "Visual Servoing" [Hashimoto93] regroupe une précieuse collection d'articles du domaine. Ce livre tente de présenter un éventail de l'état de l'art en asservissement visuel et inclut des articles utilisant différentes approches et stratégies de commande, dont des techniques de commande linéaire, non linéaire [Hashimoto93], adaptative, ou basées sur la logique floue et les réseaux de neurones [Suh93]. De même, les différents chapitres couvrent des applications diverses des techniques d'asservissement visuel, allant de l'assemblage de pièces [Nelson et al.93b], à la préhension d'objets en mouvement [Allen et al.93]. On y retrouve les travaux de Feddema [Feddema et al.93] et Hashimoto [Hashimoto et al.93], ainsi que des articles sur le suivi d'objet en mouvement que nous détaillerons dans la prochaine section.

Concernant l'utilisation de caméras déportées, qui aboutissent par nature à des systèmes différents des systèmes embarqués ("eye-in-hand") que nous avons décrits précédemment, on peut citer les travaux récents de [Dornaika95] (qui nécessitent une calibration fine pour localiser l'objet d'intérêt par rapport à l'effecteur) et [Hager94]

Tâches de suivi 13

(qui utilise un système stéréoscopique non calibré).

1.2 Tâches de suivi

Concernant les problèmes de suivi d'objet mobiles, Papanikolopoulos et al. [Papanikolopoulos et al.93] utilisent des techniques classiques de l'automatique théorique (commande optimale LQG, placement de pôle). Cependant, le mouvement de l'objet est considéré dans leur méthode comme une perturbation négligeable, ce qui entraîne des erreurs de traînage au cours du suivi.

Nelson et al. [Nelson et al.93a] prennent en compte les singularités et les butées mécaniques du robot afin de les éviter et d'agrandir l'espace des configurations de la caméra où le suivi reste possible. Ici encore, le mouvement de l'objet n'est pas estimé ou utilisé dans la commande.

Allen et al. [Allen et al.93] décrivent un système pour la préhension dynamique d'objets en mouvement. Le système est constitué d'une paire de caméras fixes pour suivre en temps réel le mouvement plan d'un objet (en l'occurence un train électrique) et utilise cette information pour commander un bras manipulateur pour saisir dynamiquement l'objet en mouvement. Cette application intéressante réunit la poursuite visuelle en temps réel et la préhension stable et dynamique d'objets en mouvement. Ils ont développé un algorithme d'estimation du mouvement de l'objet afin de réduire le problème des erreurs de traînage. Cet algorithme, basé sur l'utilisation de filtres à coefficients constants $(\alpha-\beta-\gamma)$, a le mérite d'être simple à mettre en œuvre mais manque cependant d'efficacité pour des mouvements complexes.

Par ailleurs, Corke et al. [Corke et al.92], [Corke94] se sont intéressés aux facteurs dont les effets limitent les performances des systèmes d'asservissement visuel. Une analyse des effets dynamiques (temps de retard dans la perception et la commande, variations dûes aux effets de perspective, résonance mécanique) dans le cas de la poursuite d'un objet est réalisée.

On peut également citer les travaux menés en vision animée utilisant comme système de perception/action des têtes binoculaires actives. Le concept de la vision animée ("animate vision") dûe à Ballard [Ballard91] est basé sur une analogie avec le système visuel humain. Les têtes stéréoscopiques actives permettent en effet de réaliser des tâches de fixation ou de focalisation d'attention ("gaze control"). Parmi les travaux dans ce domaine, on peut citer

[Brown92], [Milios et al.93], [Tsosos92], [Viéville et al.95], [Murray et al.95], [Bobet94] et [Brunnström et al.96]. Genéralement, ces travaux portent surtout sur l'estimation de la position 2D puis 3D du point choisi pour la fixation, et s'intéressent en fait assez peu aux lois de commande en tant que telles. Dans [Crowley et al.95] et [Mesrabi94] cependant, une comparaison entre une commande en position dans l'espace 3D et une commande par asservissement visuel 2D est effectuée, aboutissant à un avantage indéniable de l'asservissement visuel en cas de perturbations dans le système (erreurs de calibration par exemple).

Signalons enfin les travaux menés à l'ENSPS Strasbourg, où les aspects de positionnement et de suivi d'objets sont réalisés en utilisant une base de données représentant les objets pouvant se trouver dans la scène [Doignon et al.94], [Doignon et al.95]. L'ambiguïté échelle/profondeur est levée à partir d'un seul point de vue par l'utilisation de la contrainte de rigidité des objets. Ceci suppose bien sûr un a priori sur l'environnement dans lequel agit le robot.

Mis à part la modélisation particulière, le type de commande ou l'approche architecturale du problème d'asservissement visuel, toutes les recherches menées présentent des résultats significatifs dans le sens de l'intégration en temps réel de la vision et de la manipulation en robotique. Finalement, l'ensemble de ces travaux incluent une validation expérimentale des résultats théoriques.

1.3 L'approche fonction de tâche

L'approche fonction de tâche a été développée par C. Samson, M. Le Borgne et B. Espiau [Samson et al.91] pour la commande des robots manipulateurs. Cette approche a servi de base pour développer la commande référencée capteurs, et F. Chaumette [Chaumette90] l'a appliquée au cas particulier du capteur vision. Dans ce contexte, il a généralisé le calcul des matrices d'interactions à plusieurs types de primitives géométriques. Il décrit aussi un cadre pour la conception et l'exécution des tâches de positionnement en utilisant la vision, basé sur un formalisme emprunté à la théorie des mécanismes. Il introduit le concept de "liaisons virtuelles" comme un moyen de décrire les tâches référencées capteurs. En fait, les tâches sont exprimées comme des ensembles de contraintes sur le mouvement d'un repère attaché à la caméra. A la suite de ces travaux, les étudiants de P. Rives à Sophia Antipolis se sont intéréssés à l'utilisation de la commande référencée vision pour les robots mobiles [Pissard-Gibollet93] [Rives et al.92], ainsi qu'à la détection des singularités de la matrice d'interaction dans le cas de trois points [Michel et al.93]. B. Espiau à

Grenoble a mené des recherches sur la robustesse de l'asservissement aux erreurs de calibration [Espiau93] et sur le contrôle du zoom d'une caméra [Espiau95]. Parallèlement, F. Chaumette a poursuivi ses recherches dans le domaine de l'asservissement visuel en traitant le cas de la poursuite d'un objet mobile. Le mouvement de l'objet est compensé en intégrant dans la loi de commande, soit un terme intégrateur [Chaumette et al.91] (ce qui permet de ne suivre correctement que des objets à vitesse constante), soit une estimation de ce mouvement [Chaumette et al.93] basée sur un filtrage de Kalman. Cette dernière méthode ne permet pas de considérer des ruptures de modèle dans le mouvement de l'objet (changements brusques de direction ou d'amplitude par exemple). Il s'est intéréssé aussi à l'asservissement visuel basé sur des informations non plus de position dans l'image mais de mouvement (en l'occurrence les paramètres affines du mouvement 2D) pour réaliser une tâche d'alignement [Sundareswaran et al.94] [Sundareswaran et al.96]. L'intérêt de cette approche réside dans le fait qu'elle ne nécessite pas d'extraction de primitives caractéristiques dans l'image et qu'elle est donc indépendante des objets constituants la scène. Signalons enfin les travaux des chercheurs du LASMEA à Clermont-Ferrand sur le couplage caméra/laser [Urban90] [Motyl92] [Urban et al.94] [Khadraoui et al.95]. Leur contribution a consisté essentiellement dans le calcul des matrices d'interaction associées à ce capteur particulier.

Chapitre 2

Schéma de commande

Nous présentons au début de ce chapitre le formalisme de l'approche fonction de tâche dans le contexte particulier de la commande référencée vision. Nous décrivons ensuite les expérimentations réalisées pour valider les méthodes et algorithmes développés dans cette thèse. Enfin, nous présentons une technique de réglage du gain proportionnel adaptatif de la commande.

2.1 La matrice d'interaction

La vitesse des informations visuelles \underline{s} (par exemple, les coordonnées d'un point ou les paramètres dans l'image d'une droite) dans l'image peut s'écrire [Chaumette90], [Espiau et al.92] :

$$\underline{\dot{s}} = L_{\underline{s}}^T T \tag{2.1}$$

οù

- $T=(V,\Omega)$ est le vecteur vitesse représentant le mouvement relatif entre la caméra et la scène. $V=(V_x,V_y,V_z)$ et $\Omega=(\Omega_x,\Omega_y,\Omega_z)$ sont respectivement les composantes de translation et de rotation de T dans le repère de la caméra ;
- L_s^T est une matrice, appelée matrice d'interaction, qui caractérise complètement l'interaction entre le capteur et son environnement. Cette matrice, équivalente à une matrice jacobienne, peut être calculée explicitement pour les paramètres représentant la projection dans l'image des primitives géométriques paramétrables telles que, par exemple, les points, droites, cercles, sphères et cylindres.

Nous rappelons ici le calcul classique de la matrice d'interaction dans le cas du point.

Sans perte de généralité, nous supposons la distance focale de la caméra égale à 1 de telle sorte qu'un point de coordonnées $\underline{x} = (x, y, z)^T$ dans le repère caméra se projette dans le plan image sous la forme d'un point de coordonnées $\underline{X} = (X, Y, 1)^T$ avec (voir Figure 2.1) :

$$\underline{X} = \frac{1}{z} \underline{x}$$

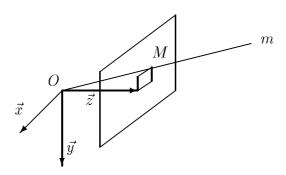


Figure 2.1: Projection perspective d'un point

En dérivant par rapport au temps la relation précédente, nous obtenons :

$$\underline{\dot{X}} = -\frac{\dot{z}}{z^2} \underline{x} + \frac{1}{z} \underline{\dot{x}}$$

Sachant que l'équation fondamentale de la cinématique pour un objet immobile et une caméra mobile est :

$$\dot{x} = -V - \Omega \wedge x$$

Nous obtenons:

$$\underline{\dot{x}} = -\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} - \begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \wedge \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -V_x + y \Omega_z - z \Omega_y \\ -V_y + z \Omega_x - x \Omega_z \\ -V_z + x \Omega_y - y \Omega_x \end{pmatrix}$$

Nous en déduisons:

$$= \begin{pmatrix} -\frac{V_x}{z} + \frac{x}{z^2} + \frac{x}{z^2} + \frac{x}{z^2} - (1 + \frac{x^2}{z^2}) \Omega_y + \frac{y}{z} \Omega_z \\ -\frac{V_y}{z} + \frac{y}{z^2} + (1 + \frac{y^2}{z^2}) \Omega_x - \frac{x}{z} \frac{y}{z^2} - \frac{x}{z} \Omega_z \\ 0 \end{pmatrix}$$

Ce qui peut s'écrire, en remplaçant les termes $\frac{x}{z}$ et $\frac{y}{z}$ par X et Y, sous la forme :

$$\underline{\dot{X}} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{pmatrix} = \begin{pmatrix} -\frac{V_x}{z} + \frac{X V_z}{z} + X Y \Omega_x - (1 + X^2) \Omega_y + Y \Omega_z \\ -\frac{V_y}{z} + \frac{Y V_z}{z} + (1 + Y^2) \Omega_x - X Y \Omega_y - X \Omega_z \\ 0 \end{pmatrix}$$

ou encore:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & (1+Y^2) & -XY & -X \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix}$$

Nous en déduisons la forme de la matrice d'interaction L_X^T dans le cas d'un point :

$$L_{\underline{X}}^{T} = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^{2}) & Y \\ 0 & -1/z & Y/z & (1+Y^{2}) & -XY & -X \end{pmatrix}$$
 (2.2)

Une méthode générale pour calculer la matrice d'interaction associée aux paramètres représentant les primitives géométriques paramétrables est décrite dans [Chaumette90] et [Espiau et al.92]. Les résultats explicites y sont donnés pour les primitives les plus usuelles : points, segments, droites, cercles, sphères et cylindres. Le calcul de la matrice d'interaction pour ces primitives ne présente pas de difficultés particulières. Nous ne les donnons pas ici car nous nous intéresserons uniquement à des primitives de type point dans cette thèse. Cependant, précisons que les méthodes développées sont générales, et s'appliquent à toutes sortes de primitives géométriques.

2.2 L'approche fonction de tâche

On peut parfaitement intégrer les techniques d'asservissement visuel dans l'approche fonction de tâche [Samson et al.91]. En effet, les tâches référencées vision peuvent s'exprimer comme la régulation à zéro de la fonction de tâche suivante :

$$\underline{e}(\underline{r},t) = \widehat{L}_{\underline{s}}^{T^{+}} \left(\underline{s}(\underline{r},t) - \underline{s}^{*} \right) \tag{2.3}$$

οù

- <u>s</u> est le vecteur représentant l'ensemble des informations visuelles choisies pour réaliser la tâche. Il dépend de la situation <u>r</u> de la caméra et du mouvement de l'objet considéré dans la scène, représenté par la variable temporelle t.
- \underline{s}^* est la valeur que \underline{s} doit atteindre pour que la tâche soit effectivement réalisée.
- $\widehat{L_{\underline{s}}^T}^+$ est la pseudo-inverse d'une estimée de la matrice d'interaction $L_{\underline{s}}^T$. Cette matrice représente le jacobien inverse de la tâche référencée vision et permet notamment de prendre en compte dans la commande un nombre d'informations visuelles supérieur au nombre de degrés de liberté de la caméra contraints par \underline{e} . Dans le cas où la connaissance de la profondeur z est nécessaire pour le calcul de la matrice d'interaction à chaque itération, une approximation possible, si cette estimation n'est pas disponible, consiste à considérer une matrice d'interaction constante correspondant à la position finale désirée.

L'approche fonction de tâche permet également de réaliser tout un ensemble de tâches secondaires, comme des suivis de trajectoires par exemple, si le nombre de degrés de liberté contraints par la tâche référencée vision est inférieur au nombre de degrés de liberté de la caméra [Samson et al.91], [Espiau et al.92].

2.3 La loi de commande référencée vision

Le problème de la commande se ramène du point de vue de l'automatique à la régulation de la fonction de tâche \underline{e} . Cette dernière est parfaitement réalisée, si à chaque instant $t : \underline{e}(\underline{r}(t), t) = 0$.

Nous désirons que la fonction de tâche décroisse de manière exponentielle. Elle doit donc vérifier la loi suivante :

$$\underline{\dot{e}} = -\lambda \underline{e} \quad \text{avec} \quad \lambda > 0$$
 (2.4)

D'autre part, comme \underline{e} est fonction de la situation \underline{r} de la caméra et du temps t, sa dérivée s'exprime de la façon suivante :

$$\underline{\dot{e}} = \left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) T_c + \frac{\partial \underline{e}}{\partial t} \tag{2.5}$$

où T_c est le torseur cinématique de la caméra, $\left(\frac{\partial \underline{e}}{\partial t}\right)$ représente les variations de \underline{e} dûes au mouvement de l'objet et où $\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right)$ est donné par (voir équations (2.3) et (2.1)):

$$\frac{\partial \underline{e}}{\partial \underline{r}} = \widehat{L}_{\underline{s}}^{T} + L_{\underline{s}}^{T}$$

A partir des équations (2.4) et (2.5), nous obtenons l'expression de la vitesse de la caméra [Chaumette90] :

$$T_c = \left(\frac{\widehat{\partial \underline{e}}}{\partial \underline{r}}\right)^+ \left(-\lambda \,\underline{e} - \frac{\widehat{\partial \underline{e}}}{\partial t}\right) \tag{2.6}$$

où:

• $\left(\frac{\widehat{\partial \underline{e}}}{\partial r}\right)$ peut être choisie égale à la matrice unité puisque l'on a :

$$\frac{\widehat{\partial \underline{e}}}{\partial r} = \widehat{L}_{\underline{\underline{s}}}^{T} + \widehat{L}_{\underline{\underline{s}}}^{T} = \mathbb{I}_{6}$$

- .
- λ est le gain fixant la vitesse de décroissance de la fonction de tâche. Nous proposons dans la suite de ce chapitre un schéma de réglage adaptatif de ce gain de façon à minimiser le temps de convergence tout en préservant la stabilité du système;
- $\left(\frac{\partial \underline{e}}{\partial t}\right)$ est l'estimée du mouvement propre de l'objet dans l'image. Si ce terme n'est pas correctement estimé, nous pouvons observer des erreurs de traînage pour les tâches de poursuite d'un objet mobile. Nous verrons dans le chapitre 3 comment on peut déterminer un algorithme d'estimation (puis de filtrage) de ce mouvement.

En utilisant la loi de commande (2.6), le comportement de la fonction de tâche \underline{e} est donné par :

$$\underline{\dot{e}} = \left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) \left(\frac{\widehat{\partial \underline{e}}}{\partial \underline{r}}\right)^{+} \left(-\lambda \underline{e} - \frac{\widehat{\partial \underline{e}}}{\partial t}\right) + \frac{\partial \underline{e}}{\partial t}$$
(2.7)

On obtient ainsi une condition suffisante de stabilité assurant une décroissance de $\parallel \underline{e} \parallel$:

$$\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) \left(\frac{\widehat{\partial \underline{e}}}{\partial \underline{r}}\right)^{+} = \left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) > 0 \tag{2.8}$$

et on retrouve le comportement attendu $\underline{\dot{e}}=-\lambda\ \underline{e}$ sous les deux conditions suivantes :

$$\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) \left(\frac{\widehat{\partial \underline{e}}}{\partial \underline{r}}\right)^{+} = \left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) = \mathbb{I}_{6} \tag{2.9}$$

$$\frac{\widehat{\partial \underline{e}}}{\partial t} = \frac{\partial \underline{e}}{\partial t} \tag{2.10}$$

La condition (2.9) peut se réécrire $\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) = \widehat{L}_{\underline{s}}^{T^+} L_{\underline{s}}^T = \mathbb{I}_6$ qui est vérifiée si le modèle $\widehat{L}_{\underline{s}}^T$ choisi pour $L_{\underline{s}}^T$ est exact.

Dans le cas contraire, les erreurs introduites dans ce modèle entraînent des per-

Dans le cas contraire, les erreurs introduites dans ce modèle entraînent des perturbations dans le comportement exponentiel découplé de la fonction de tâche ($\underline{\dot{e}}$ sera alors de la forme $\underline{\dot{e}} = -\lambda \left(\frac{\partial \underline{e}}{\partial \underline{r}}\right) \underline{e}$), mais n'empêchent pas la convergence tant que $\frac{\partial \underline{e}}{\partial \underline{r}}$ sera positif. Par ailleurs, la condition (2.10) permet de mettre en évidence l'importance d'estimer correctement $\frac{\partial \underline{e}}{\partial t}$ afin de supprimer les erreurs de traînage. En effet, si $\frac{\widehat{\partial \underline{e}}}{\partial t} \neq \frac{\partial \underline{e}}{\partial t}$ (soit encore $\frac{\partial \underline{e}}{\partial t} - \frac{\widehat{\partial \underline{e}}}{\partial t} = \epsilon(t) \neq 0$), alors on a : $\underline{\dot{e}} = -\lambda \underline{e} + \epsilon(t)$.

Finalement, on peut noter que si la condition (2.9) n'est pas respectée, des erreurs de traînage peuvent tout de même subsister. On a alors :

$$\frac{\dot{\underline{e}}}{\partial \underline{r}} = -\lambda \left(\frac{\partial \underline{e}}{\partial \underline{r}} \right) \underline{e} + \frac{\partial \underline{e}}{\partial t} - \left(\frac{\partial \underline{e}}{\partial \underline{r}} \right) \frac{\widehat{\partial \underline{e}}}{\partial t}$$
soit:
$$\underline{\dot{e}} = -\lambda \left(\frac{\partial \underline{e}}{\partial \underline{r}} \right) \underline{e} + (\mathbb{I}_6 - \left(\frac{\partial \underline{e}}{\partial \underline{r}} \right)) \frac{\widehat{\partial \underline{e}}}{\partial t} \quad \text{si} \quad \frac{\widehat{\partial \underline{e}}}{\partial t} = \frac{\partial \underline{e}}{\partial t}$$

Nous verrons cependant dans le prochain chapitre qu'avec un modèle correct de $L_{\underline{s}}^T$ et une bonne estimation de $\frac{\partial \underline{e}}{\partial t}$, il est possible de supprimer complètement les erreurs de traînage.

La loi de commande décrite par (2.6) suppose simplement :

- que l'on peut relier la vitesse de la caméra T_c aux coordonnées articulaires \underline{q} du robot porteur de la caméra par une relation du type $\underline{\dot{q}} = J^{-1}(\underline{q}) T_c$, où J^{-1} est la matrice jacobienne inverse du système robot-caméra ;
- que ce jacobien est connu et inversible (ce qui signifie que l'on ne traversera pas les éventuelles singularités du robot). Il est cependant possible d'éviter les singularités du robot ainsi que ses butées articulaires si la tâche référencée-vision ne contraint pas l'ensemble des degrés de liberté de la caméra [Nelson et al.93a],[Rizzo95];
- que la variable de contrôle du robot est la vitesse $\underline{\dot{q}}$ de ses coordonnées articulaires, ce qui est très généralement le cas.

Bien entendu, cette approche reste parfaitement valide si l'on ne souhaite contrôler qu'un sous-ensemble des six degrés de liberté de la caméra. Par exemple, dans le cas particulier d'un robot sous-marin [Bensalah et al.93], où seuls les deux degrés de liberté de cap et de roulis sont contrôlés par asservissement visuel, ou pour une tête de télésurveillance contrôlée en panoramique et inclinaison, on aura alors $T_c = (\Omega_x, \Omega_y)^T$.

2.4 Expérimentations réalisées

Pour l'ensemble de nos expérimentations, nous avons utilisé une caméra CCD Sony embarquée sur un robot Afma à six degrés de liberté (voir Figure 2.2) opérant dans un environnement architectural décrit sur la Figure 2.3.

Afin de montrer la généralité de l'approche proposée, deux tâches de positionnement et de poursuite de cibles ont été implémentées sur ce système expérimental :

• la première consiste à contrôler les six degrés de liberté de la caméra afin de réaliser un suivi complet, en position et en orientation, entre la caméra et l'objet poursuivi animé d'une succession de mouvements à vitesse constante. L'objet considéré est constitué d'un carré. Plus précisément, les informations visuelles utilisées dans cette tâche référencée vision sont les coordonnées des quatre points représentant les extrémités du carré (voir Figure 2.4):

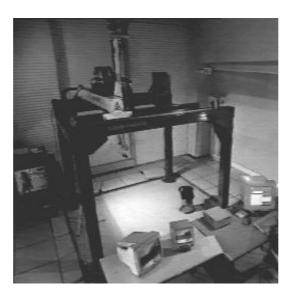


Figure 2.2: Cellule expérimentale

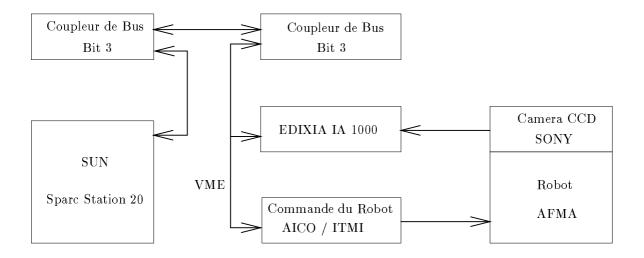


Figure 2.3 : Architecture de la cellule de vision dynamique active

 $\underline{s} = (X_1, ..., X_4, Y_1, ..., Y_4)^T$. Le calcul de la position désirée dans l'image \underline{s}^* correspondant à l'attitude désirée de la caméra par rapport au carré donne le vecteur de consigne $\underline{s}^* = (X_1^*, ..., X_4^*, Y_1^*, ..., Y_4^*)^T$.

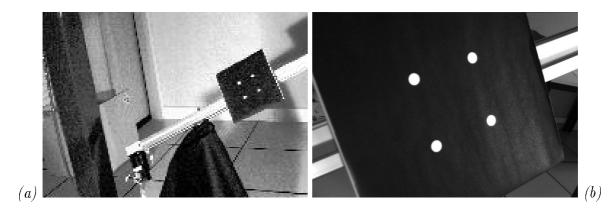


Figure 2.4 : Positions initiale et désirée du carré dans l'image

Pour le calcul de la fonction de tâche (2.3), nous utilisons à la place de $L^T_{\underline{s}}$ la matrice d'interaction $L^T_{\underline{s}|\underline{s}=\underline{s}^*}$, qui est la valeur de la matrice d'interaction correspondant à la position désirée $\underline{s}=\underline{s}^*$. Nous obtenons à partir de l'équation (2.2) :

$$L_{\underline{s}|\underline{s}=\underline{s}^*}^T = \begin{pmatrix} -1/z_1^* & 0 & X_1^*/z_1^* & X_1^*Y_1^* & -1 - X_1^{*^2} & Y_1^* \\ 0 & -1/z_1^* & Y_1^*/z_1^* & 1 + Y_1^{*^2} & -X_1^*Y_1^* & -X_1^* \\ & .. & .. & .. & .. & .. & .. \\ & -1/z_4^* & 0 & X_4^*/z_4^* & X_4^*Y_4^* & -1 - X_4^{*^2} & Y_4^* \\ 0 & -1/z_4^* & Y_4^*/z_4^* & 1 + Y_4^{*^2} & -X_4^*Y_4^* & -X_4^* \end{pmatrix}$$

et la loi de commande (2.6) s'écrit :

$$T_c = -\lambda L_{\underline{s}|\underline{s}=\underline{s}^*}^{T^+} (\underline{s} - \underline{s}^*) - \widehat{\frac{\partial \underline{e}}{\partial t}}$$

Dans ce cas, les conditions (2.8) et (2.9) de convergence et de comportement exponentiel découplé ne sont vérifiées que dans un voisinage de la position désirée. Nous verrons cependant dans les résultats expérimentaux que le comportement réel est satisfaisant, même pour des positions initiales de la caméra très éloignées de la position désirée.

• la seconde consiste à contrôler deux degrés de liberté de la caméra, à savoir les rotations autour de ses axes x (pan Ω_x) et y (tilt Ω_y), afin que le centre de gravité de l'objet poursuivi de coordonnées $\underline{s} = (X, Y)^T$ dans l'image, apparaisse en permanence au centre de l'image, de coordonnées $\underline{s}^* = (0, 0)^T$ (voir Figure 2.5). L'objet réalise divers types de mouvement (vitesse constante, accélération constante, mouvement complexe).



Figure 2.5 : Positions initiale et désirée de l'objet dans l'image

La matrice d'interaction correspondante s'obtient à partir de l'équation (2.2) et nous avons :

$$L_{\underline{s}}^{T} = \begin{pmatrix} XY & -(1+X^{2}) \\ \\ (1+Y^{2}) & -XY \end{pmatrix}$$

La loi de commande s'exprime en fonction de la fonction de tâche définie par (2.3):

$$\underline{e} = L_{\underline{s}}^{T-1} \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} Y/(1+X^2+Y^2) \\ -X/(1+X^2+Y^2) \end{pmatrix}$$

et est donnée par (voir équation (2.6)):

$$T_c = -\lambda \left(\frac{Y/(1+X^2+Y^2)}{-X/(1+X^2+Y^2)} \right) - \frac{\widehat{\partial \underline{e}}}{\partial t}$$
 (2.11)

On peut noter dans ce cas que le modèle $\widehat{L}_{\underline{s}}^T$ choisi dans la fonction de tâche est exact (aux erreurs de mesure dans le plan image près) et que l'on a $\widehat{L}_{\underline{s}}^{T^+} = L_{\underline{s}}^{T^{-1}}$. Les conditions (2.8) et (2.9) sont donc toujours vérifiées.

Grâce à une carte de traitement d'image Edixia IA 1000 et à la simplicité des scènes considérées, tous les calculs nécessaires à la réalisation d'une itération de la boucle de commande (traitement d'image, filtrage de Kalman, test du GLR et calcul de la commande) sont effectués en moins de 40 ms, ce qui fait que la cadence de la loi de commande est égale à la cadence vidéo (25 Hz). Elle est même de 50 Hz dans le cas du suivi d'un point (puisque l'on acquiert alors une trame, tout en traitant la précédente).

Nous allons à présent décrire précisément les différents étapes nécessaires à l'élaboration de la loi de commande (2.6), à savoir le réglage du gain λ (qui sera l'objet de la fin de ce chapitre) et l'estimation de $\left(\frac{\partial \underline{e}}{\partial t}\right)$ qui sera l'objet du chapitre suivant.

2.5 Réglage du gain proportionnel λ

Ce gain régit la vitesse de convergence de la fonction de tâche \underline{e} : plus une convergence rapide est souhaitée, plus ce gain doit être fixé à une valeur forte. Cependant, λ agit également directement sur la stabilité du système et une valeur trop forte peut entraîner l'instabilité et donc la divergence de la loi de commande (voir Figure 2.6). Le réglage de λ consiste donc à trouver un compromis satisfaisant entre la stabilité, qui doit absolument être assurée, et une vitesse de convergence la plus rapide possible.

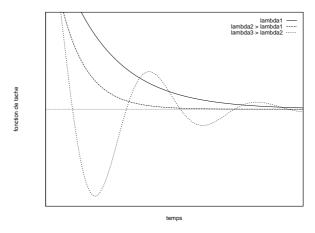


Figure 2.6 : Influence de la valeur du gain λ sur la décroissance de la fonction de tâche

Plus précisément, λ dépend essentiellement de la période d'échantillonnage du système : plus cette période est faible, plus λ peut être choisi élevé. Cette période d'échantillonnage peut être déterminée une fois pour toutes en fonction des capacités software et hardware du matériel informatique sur lequel sera exécuté le traitement d'image et le calcul de la loi de commande. Elle est de 25 et 50 Hz pour nos deux types d'expérimentations suivant qu'on s'intéresse au suivi d'un carré ou d'un point. Il s'avère également que λ dépend de la valeur de la fonction de tâche \underline{e} et donc de l'erreur dans l'image $\underline{s} - \underline{s}^*$. En effet, pour une forte valeur de \underline{e} , λ devra être choisi faible (toujours en raison des problèmes de stabilité) car la consigne de commande du robot exprimée en vitesse de déplacement dépend du terme $\lambda \underline{e}$ (voir équation (2.6); par contre, pour de faibles valeurs de \underline{e} , λ pourra prendre des valeurs plus fortes. En raison de la loi d'évolution exponentielle choisie pour <u>e</u> et qui a permis d'élaborer la loi de commande (2.6), une valeur de λ constante, préservant la stabilité du système, va alors donner le comportement suivant : une forte décroissance aux premières itérations, puis une décroissance extrêmement faible pour arriver à la convergence. Une valeur plus forte de λ entrainerait une décroissance plus rapide en fin de convergence, mais ne préserverait alors sans doute plus la stabilité pour de fortes valeurs de \underline{e} .

Signalons que les expérimentations considérées pour valider les méthodes de réglage du gain λ et qui seront décrites dans la suite de ce chapitre concernent des tâches de positionnement par rapport à un point et par rapport à un carré qui sont fixes. En effet seul le comportement en phase de convergence nous intéresse ici.

2.5.1 Génération de trajectoire dans l'image

Une solution pour optimiser la vitesse de convergence tout en préservant la stabilité du système consiste, quand cela est possible, à contrôler la trajectoire dans l'image que doit suivre \underline{s} pour atteindre \underline{s}^* . Plus précisément, à chaque itération k de la boucle de commande est calculée une valeur $\underline{s}^*_{(k)}$ de manière à ce que l'erreur $\underline{e}_{(k)} = \widehat{L}_{\underline{s}}^{T^+}(\underline{s}_{(k)} - \underline{s}_{(k)}^*)$ soit toujours inférieure à une certaine valeur e_{max} (ce qui revient à saturer la commande). Cela nous permet d'utiliser un gain constant λ , déterminé pour assurer la convergence le plus rapidement possible à partir d'une erreur initiale e_{max} [Spindler93].

Cette solution a été employée pour une tâche qui consiste à positionner la caméra par asservissement visuel, de telle sorte que le centre de gravité de l'image d'un objet initialement à la position $\underline{s} = (X_i, Y_i)^T$ se retrouve à la position $\underline{s}^* = (X_d, Y_d)^T$ dans l'image (voir Figure 2.7), en controlant uniquement les composantes de vitesse d'orientation Ω_x et Ω_y de T_c :

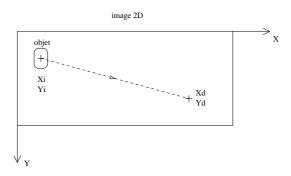


Figure 2.7: Positionnement par rapport à un point

Les courbes de la Figure 2.8 traduisent le comportement en fonction du temps des composantes $s_i - s_i^*$, (dans notre cas les coordonnées $(X - X_d, Y - Y_d)$ en pixels) à gauche et les deux composantes de T_c à droite. On a utilisé ici un gain λ constant et l'on peut observer la décroissance exponentielle de $s - s^*$ et T_c .

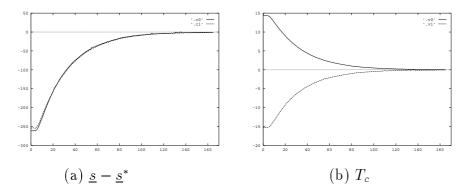


Figure 2.8 : Positionnement par rapport à un point : $\lambda = 1, 5$

Pour cette tâche, puisqu'il nous est possible de connaître la position initiale du centre de gravité et sa position d'arrivée désirée dans l'image, par le simple calcul de la droite passant par ces deux points, il nous est possible de déterminer les coordonnées de points intermédiaires $\{Xd_j, \ldots, Yd_j, \ldots\}$ situés sur cette droite et espacés de façon régulière dans l'image (voir Figure 2.9) :

Ainsi pour aller à la position (X_d, Y_d) , la caméra cherchera à amener l'image de l'objet sur les différentes positions intermédiaires successives. Comme la distance entre deux points intermédiaires est plus faible que la distance totale à parcourir, l'écart entre la position relevée à chaque itération et le point intermédiaire suivant sera réduit, de sorte que l'on pourra autoriser une commande plus forte, par l'utilisation d'un gain λ plus élevé.

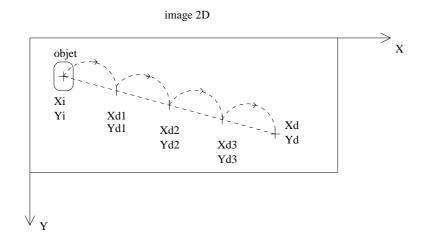


Figure 2.9 : Utilisation de points intermédiaires

La Figure 2.10 a été obtenue en utilisant 4 points intermédiaires. Avec les mêmes conditions initiales que précédemment, la convergence est réalisée en environ 2 secondes. Le temps de convergence est amélioré par l'introduction d'un gain plus élevé, déterminé de manière très simple, mais cette méthode a l'inconvénient de présenter différentes phases d'accélération et de décélération du mouvement du robot à l'approche de chaque point intermédiaire. Pour pallier à cet inconvénient et obtenir un temps de convergence encore plus rapide, il est possible de générer s^* à chaque itération, ce qui fait disparaître le phénomène d'accélérations / décélérations. Sur les résultats obtenus (voir Figure 2.11), on peut observer que $s(t) - s^*(t)$ est à peu près constant tant que $s^*(t)$ varie (ce qui entraîne une commande également constante pendant cette phase). Ensuite, dès que $s^*(t)$ devient constant (et égale à la valeur finale désirée), le comportement exponentiel décroissant apparaît de nouveau. Précisons que le palier de commande visible aux environs de l'itération 50 correspond à une réduction du gain de moitié à proximité de la position désirée.

En réalité, cette méthode souffre surtout de ne pas être généralisable à toutes sortes de tâches. La difficulté est en fait reportée sur la construction de la trajectoire $\underline{s}_{(k)}^*$, construction qui peut s'avérer très délicate, voire impossible, notamment lorsque la dimension de \underline{s} est supérieure au nombre de degrés de liberté contrôlés par la tâche-référencée-vision. En effet, seules quelques trajectoires définies dans l'image sont alors physiquement réalisables.

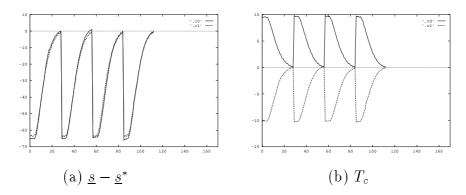


Figure 2.10 : Utilisation de 4 points intermédiaires : $\lambda=4$

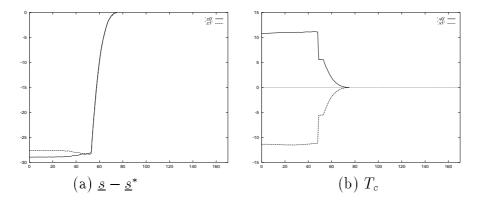


Figure 2.11 : Utilisation d'une consigne s^* variable dans le temps : $\lambda=8$

2.5.2 Gain adaptatif

Afin d'optimiser la vitesse de convergence, notamment pour les dernières itérations, une autre solution consiste à régler λ de manière adaptative en fonction de la valeur courante de \underline{e} . Les travaux décrits dans [Santos et al.92] ont montré expérimentalement que λ pouvait suivre l'évolution suivante :

$$\lambda_k = \alpha exp^{(-\beta|e_{(k)}|_{max})} + \gamma \text{ avec } \alpha, \beta, \gamma > 0$$
 (2.12)

où $|e_k|_{max}$ est, à l'itération k, la valeur absolue de la composante de \underline{e} d'amplitude la plus forte. Le réglage des valeurs de α , β et γ peut-être réalisé en choisissant ces valeurs à partir de la courbe expérimentale que nous pouvons déterminer par une série de réglages de λ pour différentes valeurs initiales de \underline{e} . Il s'agit de trouver la courbe théorique qui s'approche le plus des points expérimentaux suivant un critère des moindres carrées (voir Figure 2.12). Comme la fonction dépend de trois paramètres, trois points de mesure au moins sont nécessaires (on en choisit en pratique davantage pour obtenir une solution plus robuste). Cette méthode s'avère donc lourde et fastidieuse.

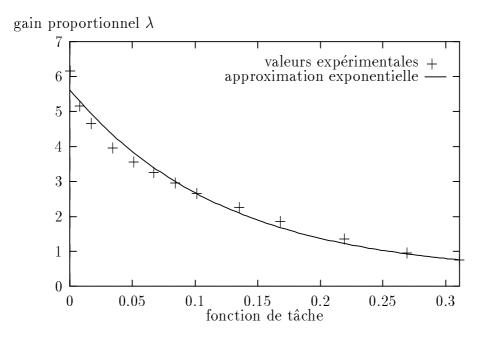


Figure 2.12: Approximation du gain proportionnel λ par une loi exponentielle

La seconde solution que nous avons développée pour déterminer α , β et γ consiste à se fixer une contrainte sur la commande : pour des raisons mécaniques (pas de vitesses trop importantes) ou liées au traitement d'image (afin de ne pas avoir un

déplacement trop important entre deux images acquises successivement), nous cherchons à limiter la valeur de la commande T_c afin qu'elle ne dépasse pas une valeur seuil T_{cmax} . Nous souhaitons donc avoir à chaque itération k:

$$\mid T_{c_{(k)}} \mid \leq T_{cmax} \tag{2.13}$$

soit:

$$\lambda_{(k)} |e_{(k)}| = (\alpha exp^{(-\beta|e_{(k)}|)} + \gamma) |e_{(k)}| \le T_{cmax}$$
 (2.14)

Nous pouvons considérer que le mouvement propre de l'objet est nul dans l'équation ci-dessus $\left(\frac{\partial e}{\partial t}\right) = 0$, car on ne s'intéresse ici qu'au réglage de la vitesse de convergence.

Notons $f_{(e)} = (\alpha exp^{(-\beta|e|)} + \gamma) |e|$. Nous cherchons la valeur de |e| correspondant au passage par une valeur maximale localement de $f_{(e)}$ (et donc à une valeur maximale de la loi de commande choisie égale à T_cmax). La dérivée de f par rapport à |e| a pour équation :

$$f'_{(e)} = \alpha (1 - \beta |e|) exp^{(-\beta |e|)} + \gamma$$
 (2.15)

Cette dérivée s'annule pour la valeur de |e| suivante, notée e_{Tcmax} :

$$e_{Tcmax} = \frac{1 - W(\frac{-\gamma exp^1}{\alpha})}{\beta} \tag{2.16}$$

où W(x) est solution de l'équation transcendentale W(x) $exp^{W(x)} = x$, soit $W(a) = b \Leftrightarrow b$ $exp^b = a$.

• Remarque : W, représentée figure 2.13, a pour intervalle de définition $[-exp^{-1}; +\infty]$. e_{Tcmax} sera donc défini si l'on a :

$$\frac{-\gamma exp^{1}}{\alpha} \ge -exp^{-1} \Rightarrow \frac{\gamma}{\alpha} \le exp^{-2} \simeq 0,135 \tag{2.17}$$

ce qui nous donne une première condition sur α et γ concernant l'existence de e_{Tcmax} . Si cette condition n'est pas respectée alors T_c n'admet pas de maximum local et atteint sa valeur la plus forte à l'état initial.

La valeur de f correspondant à la valeur particulière e_{Tcmax} s'écrit de la manière suivante :

$$f_{(e_{T_{cmax}})} = T_{cmax} = \frac{(1 - W(\frac{-\gamma exp^1}{\alpha}))(\alpha exp(W(\frac{-\gamma exp^1}{\alpha}) - 1) + \gamma)}{\beta}$$
(2.18)

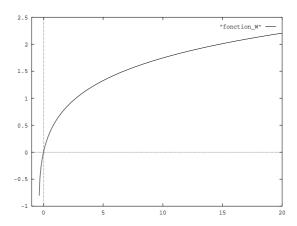


Figure 2.13 : Courbe représentative de la fonction W

que nous pouvons noter $\frac{N_{(\alpha,\gamma)}}{\beta}$. Nous pouvons donc satisfaire la condition (2.13) en faisant varier β , et β seulement, si α et γ ont préalablement été fixés. Plus précisément, on choisira :

$$\beta = \frac{N_{(\alpha,\gamma)}}{T_{cmax}} \tag{2.19}$$

 T_{cmax} est un maximum local de la commande. Pour s'assurer qu'il correspond bien à un maximum global de la commande, il faut s'assurer qu'il est plus grand que la valeur de la commande à l'état initial, calculée à partir de la valeur initiale de la fonction de tâche.

Pour déterminer α et γ , nous procédons de la manière suivante : γ est obtenu en se fixant un gain minimal, noté λ_1 , qui correspond à l'asymptote quand e tend vers l'infini de la fonction exponentielle (2.12). La valeur de λ_1 est réglée de manière à être la plus forte possible, tout en préservant la stabilité du système, pour une valeur initiale importante de e. α est ensuite déterminé à l'aide d'un second gain λ_2 obtenu par un réglage identique mais pour une valeur de e très faible ($e \simeq 0$). En utilisant (2.12), nous en déduisons :

$$\begin{cases}
\gamma = \lambda_1 \\
\alpha = \lambda_2 - \lambda_1
\end{cases}$$
(2.20)

2.5.2.1 Résultats expérimentaux

Nous présentons ici les résultats obtenus pour l'expérimentation qui consiste à positionner un point au centre de l'image. Deux cas de figure ont été considérés : gain constant ou adaptatif.

- Pour cette première expérimentation, nous avons fixé le gain λ à 1.06. Ceci est la valeur calculée du gain à l'instant initial dans l'expérimentation suivante et nous permet de comparer les performances des 2 cas. Les résultats obtenus sont décrits sur la Figure 2.14 (erreur dans l'image $(s_i s_i^*)$, composantes de la fonction de tâche e et valeurs de la commande en translation T_{ci}).
 - On observe bien une décroissance exponentielle de la fonction de tâche. La commande est maximale à l'état initial et atteint rapidement des valeurs très faibles.
- Partant de la même position initiale que précédemment, nous utilisons dans cette seconde expérimentation un gain adaptatif avec comme objectif de ne pas dépasser en commande la valeur de 10 dg/s. α et γ sont fixés à 5.28 et 0.34. Ces valeurs obéissent au souhait d'avoir un certain gain fort près de la convergence. Déterminons à présent β .

Nous avons d'après (2.16) et (2.18) :
$$e_{Tcmax} = \frac{1.22}{\beta}$$
, et : $T_{cmax} = \frac{2.32}{\beta}$

On veut $T_{cmax} = 10 \ dg/s = 0.174 \ rad/s$, ce qui donne $\beta = 13.3$. On en déduit $e_{Tcmax} = 0.092$. On passera donc bien par ce maximum puisque e_{Tcmax} est plus faible que l'erreur initiale $e_{initial} = 0.15$. On peut vérifier que ce maximum est un maximum global. On trouve : $(T_{cinitial} = 9.1 \ dg/s) < (T_{cmax} = 10 \ dg/s)$. Ces résultats se vérifient en effet aussi bien de façon théorique (voir la Figure 2.15 qui donne le comportement de T_c et λ en fonction de e pour un gain constant et pour un gain adaptatif) qu'expérimentale (voir Figure 2.16.c). On peut également observer une légère oscillation en fin d'asservissement. Par rapport à l'expérimentation précédente utilisant un gain constant, le temps de convergence a été diminué d'un facteur 3 (passant d'environ 4 s à 1.5 s), ce qui permet de montrer l'intérêt d'utiliser un gain variable, dépendant uniquement de la valeur courante de la fonction de tâche.

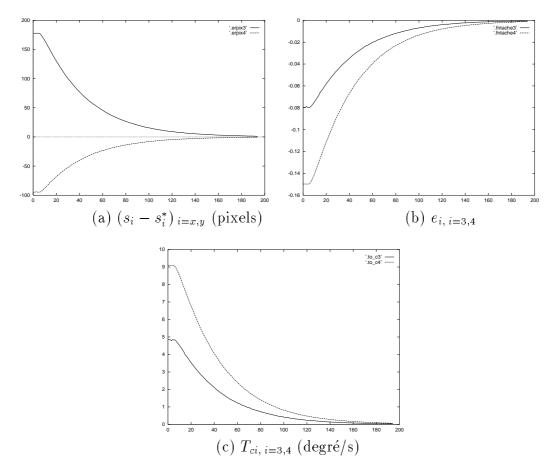


Figure 2.14 : Positionnement par rapport à un point, gain constant

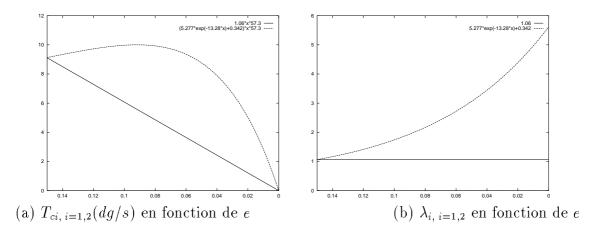


Figure 2.15 : Résultats théoriques

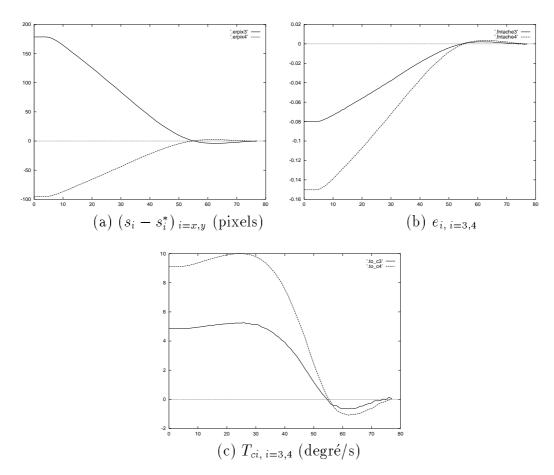


Figure 2.16 : Positionnement par rapport à un point, gain adaptatif

Nous présentons maintenant les résultats obtenus pour l'expérimentation qui consiste à se positionner par rapport à un carré à une distance et avec une orientation fixées à l'avance. Avec un gain adaptatif, deux cas de figure ont été considérés, à savoir maximum de la commande atteint initialement et maximum atteint au cours de l'asservissement. La figure 2.17 compare les comportements théoriques de la loi de commande pour ces 2 cas de figure.

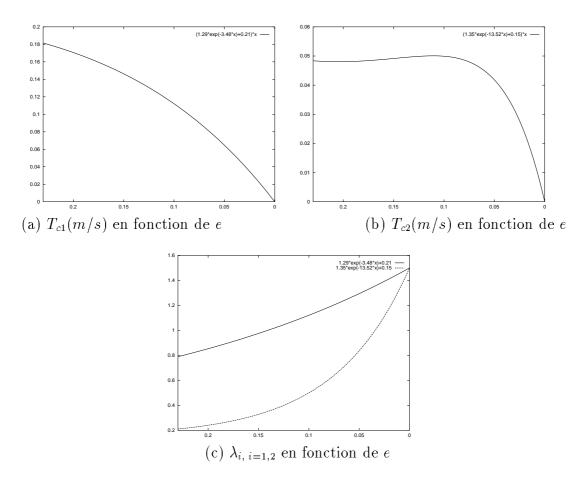


Figure 2.17 : Résultats théoriques pour les 2 cas de figure considérés (à gauche, maximum atteint à la position initiale; à droite maximum atteint pendant l'asservissement)

• Pour la première expérimentation, nous avons utilisé un gain constant égal à 0.77, gain initial dans l'expérimentation suivante. Les résultats obtenus sont visibles sur la Figure 2.18. Notons que le temps de convergence est de 13 s.

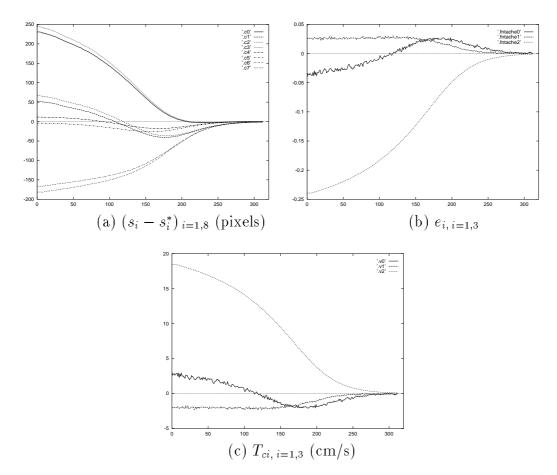


Figure 2.18 : Positionnement par rapport à un carré : gain constant

• Dans une seconde expérimentation, nous avons $\alpha = 1.29, \beta = 3.48$ et $\gamma = 0.21$. Ces valeurs ont été obtenues par la méthode expérimentale précédemment décrite (choix de α , β , γ à partir de la courbe approximant au mieux les gains réglés pour différentes positions initiales). Les résultats obtenus sont donnés sur la Figure 2.19 (erreur dans l'image $(s_i - s_i^*)$, fonctions de tâche e_i et valeurs de la commande en translation T_{ci}).

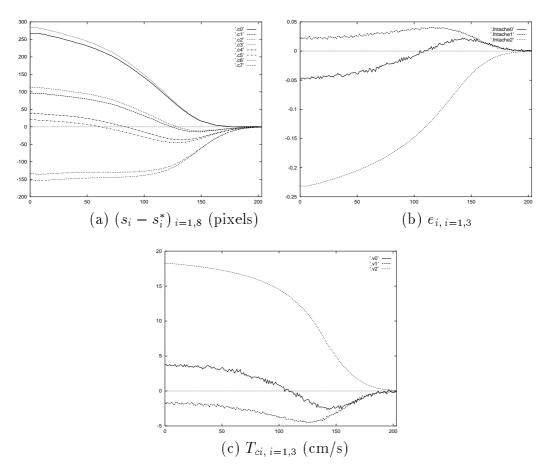


Figure 2.19 : Positionnement par rapport à un carré : coefficients du gain obtenus de manière expérimentale

Le rapport $\frac{\alpha}{\gamma}=0.16$ est supérieur à 0.135, il n'existe donc pas de maximum local de la commande (voir (2.17)). La commande est maximale à l'état initial. La plus forte valeur initiale de la fonction de tâche en valeur absolue est égale à 0.23, ce qui donne un gain λ égal à 0.79, et une commande $T_c=-\lambda e$ de 0.18 m/s. Ceci se vérifie aussi bien de façon théorique (voir Figure 2.17.a), que

de façon expérimentale (voir Figure 2.19.c). Nous observons que le temps de convergence a significativement diminué et est égal à 8 s.

Partant de la même position initiale que précédemment, nous souhaitons que la commande en translation ne dépasse pas 5 cm/s. Nous fixons α à 1.35, et γ à 0.15. Le choix des valeurs de α et γ obéit au souhait d'avoir le même gain près de la convergence que précédemment (pour e proche de 0, nous avons encore λ = α + γ = 1.5). Il nous reste à présent à déterminer β.

Nous avons d'après (2.16) et (2.18) :
$$e_{Tcmax} = \frac{1.496}{\beta}$$
, et : $T_{cmax} = \frac{0.676}{\beta}$

On veut : $T_{cmax} = 0.05 \ m/s$, ce qui donne : $\beta = 13.52$. On en déduit : $e_{Tcmax} = 0.11$. On passera donc bien par ce maximum puisque : e_{Tcmax} est inférieur à $e_{initial} = 0.23$. On peut vérifier que ce maximum est un maximum global. On a : $(T_{cinitial} = 0.048 \ m/s) < (T_{cmax} = 0.05 \ m/s)$. Ces résultats sont confirmés aussi bien sur la courbe théorique $T_c = f(e)$ (Figure 2.17.b) que sur les résultats expérimentaux (Figure 2.20.c). Le temps de convergence est de 21 s. Ceci est directement lié au choix de la commande maximale. Si on avait choisi un maximum de 20 cm/s au lieu de 5, on obtiendrait un temps de convergence équivalent à l'expérimentation précédente.

En conclusion, nous pouvons dire que la méthode développée pour le calcul théorique des différents paramètres du gain adaptatif nous semble la meilleure, car simple à mettre en œuvre et générale pour toute tâche d'asservissement visuel. Elle permet en outre un gain très appréciable en temps de convergence.

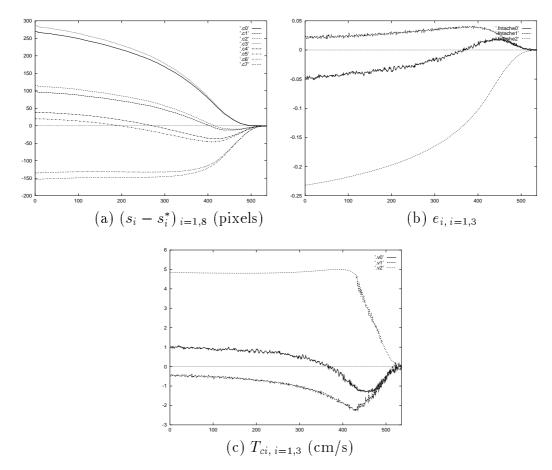


Figure 2.20 : Positionnement par rapport à un carré : coefficients du gain obtenus de manière théorique

Chapitre 3

Estimation robuste aux changements de modèle du mouvement de l'objet

Le but de ce chapitre est de présenter un estimateur de la vitesse de l'objet dans l'image, $\left(\frac{\partial \underline{e}}{\partial t}\right)$, afin de compenser dans la loi de commande les effets d'erreurs de traînage introduites par ces mouvements. Le principe employé ici consiste à utiliser les mouvements de la caméra nécessaires au suivi de l'objet afin d'en estimer le mouvement et de le compenser par des lois de commande robustes. Nous commençons par introduire un estimateur, puis son filtrage par filtre de Kalman et test du GLR (maximum de vraisemblance généralisé).

3.1 Estimation du mouvement de l'objet dans l'image

On rappelle que la loi de commande utilisée est de la forme (voir équation (2.6)) :

$$T_c = -\lambda \, \underline{e} - \frac{\widehat{\partial \underline{e}}}{\partial t}$$

et qu'il est nécessaire d'estimer correctement la contribution de la fonction de tâche dûe à un mouvement éventuel de l'objet pour supprimer les erreurs de traînage pendant la poursuite de cet objet. En effet, on a déjà vu (voir équation (2.7)) qu'avec un modèle correct de la matrice d'interaction, le comportement de la fonction de tâche s'écrit :

$$\underline{\dot{e}} = -\lambda \, \underline{e} - \frac{\widehat{\partial \underline{e}}}{\partial t} + \frac{\partial \underline{e}}{\partial t}$$

Dans [Chaumette90] et [Chaumette et al.91], l'estimation de $\left(\frac{\partial \underline{e}}{\partial t}\right)$ est réalisée par l'équation de récurrence suivante :

$$\left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right)_{(k+1)} = \left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right)_{(k)} + \mu e_k = \mu \sum_{j=0}^k e_j \text{ avec } \left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right)_0 = 0$$

où μ est un gain à régler.

On voit que cet algorithme se comporte en fait comme un intégrateur et est construit pour estimer correctement une vitesse constante de l'objet. En effet, on a :

$$\left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right)_{(k+1)} = \left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right)_{(k)} \operatorname{si} e_k = 0$$

Cet estimateur a le mérite d'être basé uniquement sur la mesure successive des valeurs de la fonction de tâche \underline{e} , qui elles-même dépendent uniquement des mesures effectuées dans l'image. Les résultats obtenus sont cependant assez peu performants en raison du modèle rudimentaire choisi. Afin de considérer des mouvements plus complexes, nous allons à présent décrire un nouvel estimateur de la vitesse de l'objet dans l'image. La mesure de cette dernière peut être obtenue à partir des mesures de la vitesse de la caméra et de la fonction de tâche. Elle est calculée en faisant la différence entre la variation totale de la fonction de tâche et la variation de la fonction de tâche dûe au mouvement de la caméra [Santos et al.92]. En effet, à partir de (2.5), nous obtenons :

$$\left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right) = \underline{\hat{e}} - \left(\frac{\widehat{\partial \underline{e}}}{\partial \underline{r}}\right) T_c$$

Après discrétisation, cette équation se réécrit :

$$\left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right)_{(k)} = \frac{\underline{e}_{(k)} - \underline{e}_{(k-1)}}{\Delta t} - \left(\frac{\widehat{\partial \underline{e}}}{\partial \underline{r}}\right)_{(k)} T_{c(k-1)}$$
(3.1)

où Δt est la période d'échantillonnage de la boucle de commande;

Comme on l'a déjà vu, $\left(\frac{\widehat{\partial \underline{e}}}{\partial \underline{r}}\right)_{(k)}$ peut être choisi comme étant égal à l'identité. La relation précédente devient donc :

$$\left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right)_{(k)} = \frac{\underline{e}_{(k)} - \underline{e}_{(k-1)}}{\Delta t} - T_{c(k-1)} \tag{3.2}$$

• Remarque: Si l'objet est supposé immobile, nous pouvons prédire la position future dans l'image des informations visuelles à partir de leur position courante et de la valeur de la commande en vitesse appliquée au robot. On obtient (voir Figure 3.1):

$$\underline{\underline{s}}_{(k)/(k-1)} = \underline{\underline{s}}_{k-1} + L_{\underline{\underline{s}}}^T T_c \Delta t \tag{3.3}$$

En utilisant (3.3) et (2.3), nous pouvons calculer une prédiction de la valeur de la prochaine fonction de tâche :

$$\underline{e}_{(k)/(k-1)} = \underline{e}_{(k-1)} + \widehat{L}_{\underline{s}(k)}^{T} \widehat{L}_{\underline{s}(k)}^{T} T_{c(k-1)} \Delta t = \underline{e}_{(k-1)} + T_{c(k-1)} \Delta t$$
 (3.4)

Notre estimateur peut donc s'écrire sous la forme (voir (3.2)) :

$$\left(\frac{\widehat{\partial \underline{e}}}{\partial t}\right)_{(k)} = \frac{\underline{e}_{(k)} - \underline{e}_{(k)/(k-1)}}{\Delta t}$$
(3.5)

qui représente (modulo Δt) l'innovation (ou l'erreur de prédiction) entre la valeur mesurée $\underline{e}_{(k)}$ de la fonction de tâche et sa prédiction. On peut noter que cette erreur de prédiction est nulle si l'objet est effectivement immobile, que cette erreur est constante si l'objet est à vitesse constante et qu'elle suit donc le modèle du mouvement de l'objet.

Revenons à présent au problème de la commande en considérant les aspects de robustesse de notre estimateur. Deux sortes d'erreurs peuvent y être introduites :

- d'une part, les erreurs dûes aux imprécisions du traitement d'image lors de l'extraction des informations visuelles \underline{s} ;
- d'autre part, les erreurs dûes à une mesure imprécise des mouvements de la caméra T_c .

L'estimée obtenue par (3.1) est donc généralement bruitée. Pour filtrer ce genre d'erreur de mesure lors d'une poursuite de cible, on rencontre dans la littérature

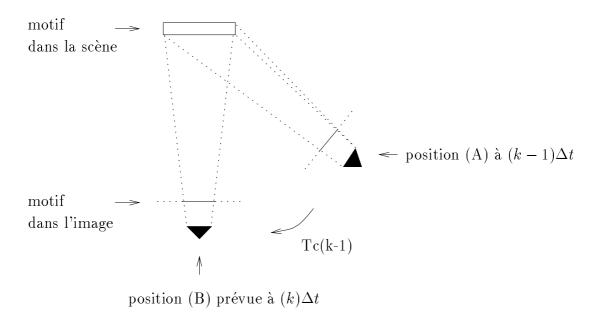


Figure 3.1 : Variation de la position de l'objet dans l'image

deux différentes approches [Blackman86], [Allen et al.91], [Hunt et al.82] : la première consiste à utiliser des filtres à coefficients constants (filtres $\alpha - \beta$ ou $\alpha - \beta - \gamma$); la seconde, basée sur le filtrage de Kalman, consiste à estimer au fur et à mesure du filtrage les coefficients du modèle de mouvement, modèle que l'on se fixe a priori. La première approche semble assez rudimentaire, notamment si l'on souhaite poursuivre un objet ayant un mouvement complexe. Aussi avons-nous choisi d'utiliser un filtre de Kalman pour robustifier notre estimateur.

Cependant, une estimation correcte n'est pas réalisée dans le cas de changements brusques au niveau du vecteur d'état. En effet, un filtre de Kalman classique, en raison de sa relative rigidité au niveau de l'évolution choisie pour le modèle d'état, considère un saut comme étant du bruit et la dynamique du modèle d'état empêche alors une estimation correcte pendant un certain nombre d'itérations nécessaires à la convergence du filtre.

La littérature propose une large gamme d'algorithmes de détection de rupture de modèle [Basseville et al.83], [Bar-Shalom et al.93]. Parmi ceux-ci, citons le test de Page-Hinkley, test de type "cusum" (somme cumulative) qui permet de détecter des sauts additifs [Hinkley71]. Ce test ne prend pas en compte les variances des erreurs d'estimation, ce qui diminue la qualité de ses performances. En effet, le

test de Page-Hinkley ne fournit pas d'estimation directe du saut, ni la variance sur l'erreur correspondante. En outre, ce test nécessite la connaissance de la moyenne initiale du signal. Nous avons donc choisi d'utiliser l'algorithme du rapport de vraisemblance généralisé (Generalized Likelihood Ratio) développé par Willsky et al. [Willsky et al.76], car cette méthode permet de détecter, d'estimer et de compenser les sauts au niveau du vecteur d'état. Cet algorithme permet donc d'obtenir une meilleure estimée en un temps moindre par rapport à un filtre de Kalman classique [Bensalah et al.94a]. En outre, cette méthode ne perturbe nullement les performances du filtrage de Kalman dans le cas d'absence de saut.

L'algorithme du GLR s'est déjà prêté avec succès à un certain nombre d'applications, parmi lesquelles la détection de manœuvres aériennes dans un but d'interception [Korn et al.82], la détection de changements au niveau de signaux géophysiques [Basseville et al.83], l'estimation du mouvement 2D en traitement de séquences d'images [Germain et al.94] ou encore la détection de discontinuités dans les courbures du bord des routes [Behringer95]. Dans [Germain et al.94] par exemple, l'algorithme du GLR permet de détecter les discontinuités du champ des vecteurs vitesses aux frontières des régions ayant des mouvements différents. Il est ainsi possible de localiser ces frontières et d'y obtenir une estimation du mouvement cohérente.

3.2 Filtrage de la mesure du mouvement de la cible

3.2.1 Cas général

Les équations de modèles d'état et de mesure du filtre de Kalman (tenant compte d'un hypothétique saut au niveau du vecteur d'état à l'instant θ) s'écrivent de manière générale sous la forme suivante :

$$x_{(k+1)} = \Phi \ x_{(k)} + v_{(k)} + \delta_{\theta,k+1} \ \nu \tag{3.6}$$

$$z_{(k+1)} = H \ x_{(k+1)} + \ w_{(k)} \tag{3.7}$$

où:

- v est le bruit d'état. Il est supposé blanc gaussien centré de matrice de covariance Q;
- $\delta_{i,j}$ est le symbole de Kronecker;
- ν est le saut hypothétique; et

• w est le bruit de mesure. Il est également supposé blanc gaussien centré de matrice de covariance R.

Les équations du filtre de Kalman sont données par :

• la prédiction du vecteur d'état :

$$x_{(k+1|k)} = \Phi \ x_{(k|k)} \tag{3.8}$$

• la matrice de covariance de l'erreur de prédiction :

$$P_{(k+1|k)} = \Phi P_{(k|k)} \Phi^T + Q$$

• l'estimation du vecteur d'état :

$$x_{(k+1|k+1)} = x_{(k+1|k)} + K_{(k+1)} \gamma_{(k+1)}$$
(3.9)

• la matrice de covariance de l'erreur d'estimation :

$$P_{(k+1|k+1)} = (\mathbb{I} - K_{(k+1)} H) P_{(k+1|k)}$$

où:

 $-\gamma_{(k+1)}$ est l'innovation entre la mesure et la prédiction précédente :

$$\gamma_{(k+1)} = z_{(k+1)} - H \ x_{(k+1|k)} \tag{3.10}$$

 $-K_{(k+1)}$ est le gain du filtre de Kalman :

$$K_{(k+1)} = P_{(k+1|k)} H^T V_{(k+1)}^{-1}$$

- $V_{(k+1)}$ est la variance de l'innovation $\gamma_{(k+1)}$:

$$V_{(k+1)} = H P_{(k+1|k)} H^T + R$$

Nous décrivons en annexe A, l'initialisation des différents éléments du filtre de Kalman pour les modèles d'état les plus classiques, c'est-à-dire en supposant que le mouvement de l'objet poursuivi est à vitesse ou à accélération constante. Le cas de modèles d'état sur lequel est rajouté un bruit corrélé y est également décrit.

Nous allons à présent considérer un filtre de Kalman avec le modèle décrit précédemment, et appliquer l'algorithme du GLR afin de détecter, estimer et compenser des changements brusques dans le vecteur d'état. Nous cherchons à détecter tout saut se produisant éventuellement à un instant inconnu θ . Afin d'éviter une banque de données croissante et trop importante, nous considérons à l'itération k une fenêtre $(k-M<\theta\leq k)$ où M est la longueur choisie de la fenêtre, et nous testons si un saut s'est produit pour tout instant θ compris dans l'intervalle de cette fenêtre. La taille de cette fenêtre M doit cependant être choisie suffisamment grande pour contrer l'aspect fortement bruité du signal à estimer.

Quand un saut se produit à l'instant θ , il influence la valeur de l'innovation et donc l'estimée de l'état. L'innovation $\gamma(k)$ peut alors être considérée comme une somme de deux termes :

$$\gamma_{(k)} = \gamma_{(k)_{nj}} + G(k; \theta) \nu \tag{3.11}$$

où:

- $\gamma_{(k)_{nj}}$ représente l'innovation s'il ne se produit pas de saut; et
- $G(k;\theta)$ ν est l'effet du saut ν qui se produit à l'itération θ , sur la valeur de l'innovation $\gamma(k)$ mesurée à l'itération k.

Le vecteur d'état s'écrit de façon similaire comme une somme de deux termes :

$$x_{(k|k)} = x_{(k|k) \ nj} + F(k;\theta) \ \nu \tag{3.12}$$

où:

- $x_{(k|k)}$ n_j est l'estimée du vecteur d'état s'il ne se produit pas de saut ; et
- $F(k;\theta)$ ν est l'effet d'un saut qui se produit à l'itération θ , sur la valeur de l'estimée du vecteur d'état à l'itération k.

Comme nous le verrons par la suite, la détection et l'estimation d'un saut sont essentiellement basées sur la valeur des matrices $G(k;\theta)$ et $F(k;\theta)$. Nous allons donc à présent les déterminer explicitement pour les différentes valeurs possibles de θ .

• Cas $\theta > k$: Pour $\theta > k$, nous avons évidemment :

$$G(k;\theta) = (0)$$

$$F(k;\theta) = (0)$$

• Cas $\theta = k$: Calculons maintenant $G(\theta; \theta)$ et $F(\theta; \theta)$. Si l'on suppose l'existence d'un saut ν à l'instant k, le vecteur d'état et la mesure peuvent s'écrire :

$$\left\{ \begin{array}{lll} x_{(\theta)} & = & x_{(\theta)\;nj} & + \nu \\ z_{(\theta)} & = & z_{(\theta)\;nj} & + H\;\nu \end{array} \right.$$

où $x_{(\theta) nj}$ et $z_{(\theta) nj}$ représentent respectivement la valeur du vecteur d'état et de la mesure en l'absence de saut. L'équation précédente et la définition de l'innovation (3.10) donnent alors :

$$\gamma_{(\theta)} = \gamma_{(\theta) \ nj} + H \ \nu \tag{3.13}$$

Par identification de cette dernière expression avec la définition de $G(k; \theta)$ donnée par (3.11), nous obtenons :

$$G(\theta;\theta) = H$$

Par ailleurs, l'équation d'estimation du vecteur d'état (3.9) et l'expression (3.13) de l'innovation $\gamma_{(\theta)}$ nous donnent :

$$x_{(\theta|\theta)} = x_{(\theta|\theta) \ nj} + K_{(\theta)} \ H \ \nu$$

Par identification avec la définition (3.12) de $F(k;\theta)$, nous en déduisons :

$$F(\theta;\theta) = K_{(\theta)} H$$

• Cas $\theta < k$: Déterminons à présent, pour tout θ comprisentre k-M et k-1, la valeur de $G(k;\theta)$ et $F(k;\theta)$. Il est possible de calculer ces termes récursivement en fonction de $G(k-1;\theta)$ et $F(k-1;\theta)$, ces valeurs étant obtenues à l'itération précédente k-1. En effet, si l'on suppose l'existence d'un saut à l'instant θ , le vecteur d'état peut s'écrire sous la forme :

$$x_{(k)} = x_{(k) nj} + \Phi^{(k-\theta)} \nu$$

En appliquant l'équation de mesure (3.7) à l'équation précédente, nous obtenons :

$$z_{(k)} = z_{(k) nj} + H \Phi^{(k-\theta)} \nu \tag{3.14}$$

Par ailleurs, en multipliant l'équation de prédiction du vecteur d'état (3.8) par H, nous avons :

$$z_{(k|k-1)} = H \ x_{(k|k-1)} = H \ \Phi \ x_{(k-1|k-1)}$$
,

et, en combinant l'équation précédente avec l'équation (3.12), nous obtenons :

$$z_{(k|k-1)} = z_{(k|k-1) \ nj} + H \Phi F(k-1;\theta) \nu \tag{3.15}$$

Nous en déduisons, en considérant les expressions (3.14) et (3.15), la valeur de l'innovation $\gamma_{(k)}$:

$$\gamma_{(k)} = \gamma_{(k) nj} + (H \Phi^{(k-\theta)} - H \Phi F(k-1;\theta)) \nu$$

Par identification avec (3.11), nous en déduisons :

$$G(k;\theta) = H \Phi^{(k-\theta)} - H \Phi F(k-1;\theta)$$

Pour calculer $F(k;\theta)$, nous combinons l'équation de prédiction du vecteur d'état (3.8) avec l'équation (3.12), ce qui donne :

$$x_{(k|k-1)} = x_{(k|k-1) nj} + \Phi F(k-1;\theta) \nu$$
(3.16)

En utilisant les expressions (3.16) et (3.11), l'équation d'estimation du vecteur d'état (3.9) s'écrit comme suit :

$$x_{(k|k)} = x_{(k|k) nj} + (\Phi F(k-1;\theta) + K_{(k)} G(k;\theta)) \nu$$

et, par identification avec (3.12), nous obtenons finalement:

$$F(k;\theta) = \Phi F(k-1;\theta) + K_{(k)} G(k;\theta)$$

Nous savons donc calculer de façon récurrente, à chaque itération k et pour tout θ compris entre k-M et k, les quantités $G(k;\theta)$ et $F(k;\theta)$. Celles-ci sont en effet nécessaires au calcul du rapport de vraisemblance permettant la détection d'un hypothétique saut sur le vecteur d'état.

Nous considérons deux hypothèses. La première est qu'aucun saut ne s'est produit, et la deuxième suppose qu'un saut d'amplitude $\hat{\nu}_{(k)}$ s'est produit à l'itération θ . Etant donné les innovations calculées $\gamma_{(k-M+1)},...,\gamma_{(k)}$, à chaque hypothèse correspond une probabilité. $\hat{\nu}_{(k)}$ est déterminé de façon à maximiser la probabilité qu'un saut s'est produit. Il représente l'amplitude du saut la plus vraisemblable à l'itération k pour chaque θ fixé si l'on considère qu'un saut s'est produit. La détection d'un saut à l'instant θ est basée sur le rapport de vraisemblance $l(k;\theta)$ donné par la formule suivante [Willsky et al.76] :

$$l(k;\theta) = 2 \log \frac{\mathcal{L}(\gamma_{(k-M+1)}, ..., \gamma_{(k)}|H_1, \nu = \hat{\nu}_{(k)})}{\mathcal{L}(\gamma_{(k-M+1)}, ..., \gamma_{(k)}|H_0)}$$

où:

- $\mathcal{L}(\gamma|H)$ représente la fonction de probabilité telle que γ vérifie l'hypothèse H;
- $\hat{\nu}_{(k)}$ est défini par :

$$\hat{\nu}_{(k)} = \operatorname{argmax} \mathcal{L}(\gamma_{(k-M+1)}, ..., \gamma_{(k)}|H_1)$$

• H_0 et H_1 sont respectivement les hypothèses qu'aucun saut ne s'est produit et qu'un saut d'amplitude $\hat{\nu}_{(k)}$ s'est produit à l'itération θ .

Après développement, ce rapport de vraisemblance peut s'écrire en fonction de $G(k; \theta)$. En effet, nous avons [Willsky et al.76]:

$$l(k;\theta) = D^{T}(k;\theta) C^{-1}(k;\theta) D(k;\theta)$$
(3.17)

avec:

$$C(k;\theta) = \sum_{j=\theta}^{k} G^{T}(j;\theta) \ V_{(j)}^{-1} \ G(j;\theta)$$
 (3.18)

et:

$$D(k;\theta) = \sum_{j=\theta}^{k} G^{T}(j;\theta) V_{(j)}^{-1} \gamma_{(j)}$$
(3.19)

Remarquons que les quantités $C(k;\theta)$ et $D(k;\theta)$ ne dépendent pas directement des valeurs $F(j;\theta)$ mais seulement des $G(j;\theta)$. Cependant $G(j;\theta)$ dépend lui de $F(j-1;\theta)$, ce qui nécessite son calcul. Signalons aussi, comme nous le verrons plus loin en (3.20) et (3.21), que la valeur de F entre dans le calcul des équations de mise à jour de l'estimée et de la variance sur l'erreur d'estimation correspondante.

Les équations (3.17), (3.18) et (3.19) montrent que $l(k;\theta)$ peut être considéré comme un test de corrélation entre les variations de l'innovation $\gamma_{(k)}$ et la signature d'un saut $G(k,\theta)$.

Pour tout θ compris entre k-M et k, nous calculons le rapport $l(k;\theta)$ et nous sélectionnons la valeur θ_m qui maximise la quantité $l(k;\theta)$ et qui représente l'instant le plus probable où un saut s'est produit. On voit bien que le test du GLR est un test à double maximisation: on maximise la probabilité qu'il y ait eu un saut en sélectionnant son amplitude (pour chaque instant probable), puis l'instant de sa production. Nous appliquons ensuite la règle de décision suivante :

$$l(k; \theta_m) \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array}$$

où ϵ est une valeur de seuil fixée. Ce seuil est facile à régler en ce qui concerne les expérimentations que nous avons réalisées. Si un saut est détecté $(l(k;\theta_m) > \epsilon)$, l'instant $\hat{\theta}$ de sa production et l'estimation $\hat{\nu}(k;\hat{\theta})$ de sa valeur sont alors donnés par :

 $\begin{cases} \hat{\theta} = \theta_m \\ \hat{\nu}(k; \hat{\theta}) = C^{-1}(k; \hat{\theta}) D(k; \hat{\theta}) \end{cases}$

Cette estimation du saut est ensuite utilisée pour mettre à jour l'estimation du vecteur d'état dans le filtre de Kalman à travers l'équation de compensation suivante [Willsky et al.76]:

$$x_{(k|k)new} = x_{(k|k)old} + [\Phi^{k-\hat{\theta}} - F(k;\hat{\theta})] \hat{\nu}(k;\hat{\theta})$$
 (3.20)

Entre la production du saut et sa détection, il s'est passé $k-\hat{\theta}$ itérations. La mise à jour de l'estimée du vecteur d'état est donc réalisée par addition de l'effet d'un saut d'amplitude $\hat{\nu}$ pendant ces $k-\hat{\theta}$ itérations suivant le modèle d'état (soit $\Phi^{k-\hat{\theta}}$ $\hat{\nu}(k;\hat{\theta})$), et en retranchant la réponse du filtre de Kalman au saut avant sa détection. Rappelons que $F(k;\hat{\theta})$ $\hat{\nu}$ représente la contribution d'un saut d'amplitude $\hat{\nu}$ se produisant à l'instant θ , à l'estimée du vecteur d'état à l'instant k.

Signalons finalement que pour prendre en compte l'erreur d'estimation du saut, il est nécessaire d'incrémenter la matrice de covariance de l'erreur d'estimation $P_{(k|k)}$. L'équation de mise à jour est déduite de l'équation précédente sachant que $C^{-1}(k;\hat{\theta})$ représente la matrice de covariance de l'erreur sur l'estimation du saut $\hat{\nu}(k;\hat{\theta})$. On obtient :

$$P_{(k|k)new} = P_{(k|k)old} + \left[\Phi^{k-\hat{\theta}} - F(k;\hat{\theta})\right] C^{-1}(k;\hat{\theta}) \left[\Phi^{k-\hat{\theta}} - F(k;\hat{\theta})\right]^{T}$$
(3.21)

La réactualisation de $P_{(k|k)}$ permet d'éviter les fausses alarmes à la suite de la détection d'un saut, car si l'on ne tient pas compte de l'erreur possible sur l'estimation du saut, on risque fortement de détecter un faux deuxième saut, voire plusieurs à la suite, ce qui entraineraît une instabilité dans l'estimation.

3.2.2 Filtre de Kalman à vitesse constante, saut en vitesse

Nous allons à présent appliquer l'algorithme du GLR décrit ci-dessus à notre cas particulier de poursuite de cible par asservissement visuel. Il s'agit de détecter des changements brusques dans la vitesse de l'objet poursuivi [Bensalah et al.94b] [Bensalah et al.95b]. Nous verrons dans les paragraphes suivants qu'il est également possible de détecter des ruptures dans l'accélération de l'objet, ou de détecter simultanément des sauts en vitesse et en accélération.

Les composantes du mouvement de l'objet n'étant pas corrélées, nous pouvons utiliser un filtre de Kalman pour chaque composante $\left(\frac{\widehat{\partial e}}{\partial t}\right)$ de $\left(\frac{\widehat{\partial e}}{\partial t}\right)$ (voir équation (2.6)). Nous choisissons ici un modèle d'état à vitesse constante et bruit corrélé (au lieu d'un simple bruit blanc) afin de prendre en compte dans le filtre de Kalman de légères variations possibles sur la vitesse de l'objet [Hunt et al.82]. En parallèle à l'exécution du filtre de Kalman, un test du GLR est effectué pour détecter, estimer et compenser un éventuel saut en vitesse dans le modèle du mouvement de l'objet. Dans ce cas précis, les équations de modèle d'état et de mesure du filtre de Kalman sont données par :

$$\begin{pmatrix} \begin{pmatrix} \frac{\partial e}{\partial t} \end{pmatrix}_{(k+1)} \\ \eta_{(k+1)} \end{pmatrix} = \Phi \begin{pmatrix} \begin{pmatrix} \frac{\partial e}{\partial t} \end{pmatrix}_{(k)} \\ \eta_{(k)} \end{pmatrix} + \begin{pmatrix} 0 \\ v_{(k)} \end{pmatrix} + \delta_{\theta,k+1} \begin{pmatrix} \alpha \\ 0 \end{pmatrix}$$
(3.22)

et:

$$\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k+1)} = H\left(\begin{array}{c} \left(\frac{\partial e}{\partial t}\right)_{(k+1)} \\ \eta_{(k+1)} \end{array}\right) + w_{(k+1)}$$
(3.23)

avec:

$$\Phi = \left(\begin{array}{cc} 1 & 1 \\ 0 & \rho \end{array}\right) \quad , \quad H = \left(\begin{array}{cc} 1 & 0 \end{array}\right)$$

et où:

- ρ est le degré de corrélation entre les accélérations successives de l'objet et peut varier entre 0 et 1 (0.3 dans toutes les expérimentations décrites à la fin de ce chapitre);
- v est le bruit d'état, supposé blanc gaussien centré et de variance q;
- α est l'amplitude d'un saut hypothétique en vitesse. Nous pouvons remarquer que la deuxième composante de la forme d'état du saut est nulle, car nous ne cherchons pas à détecter des sauts au niveau du bruit corrélé.
- $\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k)}$ est la vitesse de la cible mesurée à l'itération k (donnée par l'équation (3.1)); et
- w est le bruit de mesure, supposé blanc gaussien centré et de variance r.

Les équations du filtre de Kalman sont à présent données par :

• la prédiction du vecteur d'état :

$$\begin{pmatrix} \begin{pmatrix} \widehat{\frac{\partial e}{\partial t}} \end{pmatrix}_{(k+1|k)} \\ \widehat{\eta}_{(k+1|k)} \end{pmatrix} = \Phi \begin{pmatrix} \begin{pmatrix} \widehat{\frac{\partial e}{\partial t}} \end{pmatrix}_{(k|k)} \\ \widehat{\eta}_{(k|k)} \end{pmatrix}$$

• la matrice de covariance de l'erreur de prédiction :

$$P_{(k+1|k)} = \Phi P_{(k|k)} \Phi^T + \begin{pmatrix} 0 & 0 \\ 0 & q \end{pmatrix}$$

• l'estimation du vecteur d'état :

$$\begin{pmatrix}
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k+1|k+1)} \\
\widehat{\eta}_{(k+1|k+1)}
\end{pmatrix} = \begin{pmatrix}
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k+1|k)} \\
\widehat{\eta}_{(k+1|k)}
\end{pmatrix} + K_{(k+1)} \gamma_{(k+1)} \tag{3.24}$$

• la matrice de covariance de l'erreur d'estimation :

$$P_{(k+1|k+1)} = (\mathbb{I}_2 - K_{(k+1)} H) P_{(k+1|k)}$$

où:

- $\gamma_{(k+1)}$ est l'innovation entre la mesure et la prédiction précédente :

$$\gamma_{(k+1)} = \left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k+1)} - \left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k+1|k)}$$
(3.25)

 $-K_{(k+1)}$ est le gain du filtre de Kalman :

$$K_{(k+1)} = \frac{1}{v_{(k+1)}} P_{(k+1|k)} H^T$$

- $v_{(k+1)}$ est la variance de l'innovation $\gamma_{(k+1)}$:

$$v_{(k+1)} = H P_{(k+1|k)} H^T + r$$

Nous allons à présent appliquer la méthode du GLR décrite précédemment à notre cas particulier où l'on souhaite détecter des sauts en vitesse de la cible. Puisque le saut a une direction particulière $a=\begin{pmatrix}1\\0\end{pmatrix}$ dans l'espace d'état, l'innovation s'écrit à présent :

$$\gamma_{(k)} = \gamma_{(k)_{nj}} + g(k;\theta) \alpha \tag{3.26}$$

où $g(k;\theta)$ est un scalaire. Le vecteur d'état s'écrit de façon similaire comme une somme de deux termes :

$$\begin{pmatrix}
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)} \\
\widehat{\eta}_{(k|k)}
\end{pmatrix} = \begin{pmatrix}
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)} \\
\widehat{\eta}_{(k|k)}
\end{pmatrix}_{nj} + \alpha \underline{f}(k;\theta) \tag{3.27}$$

où $\underline{f}(k;\theta)$ est un vecteur de dimension 2. $g(k;\theta)$ et $\underline{f}(k;\theta)$ sont respectivement le produit de $G(k;\theta)$ et $F(k;\theta)$ calculés précédemment par la direction du saut a. Nous obtenons :

$$\begin{cases} g(\theta; \theta) = G(\theta; \theta) \ a = H \ a = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1 \\ \underline{f}(\theta; \theta) = F(\theta; \theta) \ a = K_{(\theta)} \ H \ a = K_{(\theta)} \end{cases}$$

et, pour $\theta < k$:

$$\begin{cases} g(k;\theta) = G(k;\theta) \ a = H \ \Phi^{(k-\theta)} \ a - H \ \Phi \ F(k-1;\theta) \ a \\ \underline{f}(k;\theta) = F(k;\theta) \ a = \Phi \ F(k-1;\theta) \ a + K_{(k)} \ G(k;\theta) \ a \end{cases}$$

soit:

$$\left\{ \begin{array}{l} g(k;\theta) = \left(\begin{array}{cc} 1 & 0 \end{array} \right) \left(\begin{array}{cc} 1 & \star \\ \star & \star \end{array} \right) \left(\begin{array}{cc} 1 \\ 0 \end{array} \right) - \left(\begin{array}{cc} 1 & 0 \end{array} \right) \left(\begin{array}{cc} 1 & 1 \\ 0 & \rho \end{array} \right) \underline{f}(k-1;\theta) \\ \underline{f}(k;\theta) = \Phi \ \underline{f}(k-1;\theta) + K_{(k)} \ g(k;\theta) \end{array} \right.$$

soit finalement:

$$\begin{cases} g(k;\theta) = 1 - \begin{pmatrix} 1 & 0 \end{pmatrix} \underline{f}(k-1;\theta) \\ \underline{f}(k;\theta) = \Phi \underline{f}(k-1;\theta) + K_{(k)} g(k;\theta) \end{cases}$$

Le rapport de vraisemblance défini par l'équation (3.17) peut s'écrire à présent sous la forme scalaire suivante :

$$l(k;\theta) = \frac{d^2(k;\theta)}{c(k;\theta)}$$

avec (voir équation (3.18)):

$$c(k;\theta) = \sum_{i=\theta}^{k} \frac{g^{2}(j;\theta)}{v_{(j)}}$$

et (voir équation (3.19)) :

$$d(k;\theta) = \sum_{j=\theta}^{k} \frac{g(j;\theta) \gamma_{(j)}}{v_{(j)}}$$

Si un saut est détecté $(l(k; \theta_m) > \epsilon$ où ϵ est une valeur de seuil fixée, et où θ_m est la valeur qui maximise $l(k; \theta)$, l'estimation de l'instant de sa production est donnée par $\hat{\theta} = \theta_m$ et son amplitude par :

$$\hat{\alpha}(k; \hat{\theta}) = \frac{d(k; \hat{\theta})}{c(k; \hat{\theta})}$$

L'équation de compensation utilisée dans le filtre de Kalman pour réinitialiser le vecteur d'état s'écrit finalement :

$$\begin{pmatrix} \left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)} \\ \widehat{\eta}_{(k|k)} \end{pmatrix}_{new} = \begin{pmatrix} \left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)} \\ \widehat{\eta}_{(k|k)} \end{pmatrix}_{old} + \begin{pmatrix} \widehat{\alpha}(k;\hat{\theta}) \\ 0 \end{pmatrix} - \widehat{\alpha}(k;\hat{\theta}) \underline{f}(k;\hat{\theta})$$

La compensation se fait par ajout, à la dernière estimation du vecteur d'état, de l'estimée du saut sous forme vectorielle, et par soustraction de la réponse du filtre de Kalman au saut entre l'instant de sa production θ et celui de sa détection k. Dans ce cas particulier et pour les expérimentations décrites à la fin de ce chapitre, la mise à jour de la matrice de covariance de l'erreur d'estimation n'a pas été faite en utilisant l'équation (3.21) mais est simplement effectuée en la réinitialisant :

$$P_{(k|k)new} = P_{(0|0)}$$

Cette procédure, immédiate à mettre en œuvre, a en effet donné des résultats expérimentaux satisfaisants.

3.2.3 Filtre de Kalman à accélération constante, saut en accélération

Dans ce paragraphe, nous appliquons l'algorithme du GLR à un autre cas particulier. Il s'agit de détecter des sauts en accélération pour un modèle d'état à accélération constante et bruit corrélé [Bensalah et al.95a]. Les nouvelles équations de modèle d'état et de mesure du filtre de Kalman sont données par :

$$\begin{pmatrix}
\left(\frac{\partial e}{\partial t}\right)_{(k+1)} \\
\eta_{(k+1)} \\
\left(\frac{\partial e}{\partial t}\right)_{(k+1)}
\end{pmatrix} = \Phi \begin{pmatrix}
\left(\frac{\partial e}{\partial t}\right)_{(k)} \\
\eta_{(k)} \\
\left(\frac{\partial e}{\partial t}\right)_{(k)}
\end{pmatrix} + \begin{pmatrix}
0 \\
v_{1(k)} \\
v_{2(k)}
\end{pmatrix} + \delta_{\theta,k+1} \begin{pmatrix}
0 \\
0 \\
\alpha
\end{pmatrix}$$
(3.28)

et:

$$\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k+1)} = H \begin{pmatrix} \left(\frac{\partial e}{\partial t}\right)_{(k+1)} \\ \eta_{(k+1)} \\ \left(\frac{\partial e}{\partial t}\right)_{(k+1)} \end{pmatrix} + w_{(k+1)} \tag{3.29}$$

avec:
$$\Phi = \begin{pmatrix} 1 & 1 & \Delta t \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{pmatrix} , \quad H = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

où Δt est la période d'échantillonnage de la boucle de commande.

En parallèle au filtre de Kalman développé sur la base du modèle ci-dessus, nous avons appliqué le test du GLR au cas particulier où l'on souhaite détecter des sauts

en accélération de l'objet. Ici, la direction du saut dans l'espace d'état est $a = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$.

L'innovation s'écrit donc de nouveau :

$$\gamma_{(k)} = \gamma_{(k)_{ni}} + g(k;\theta) \alpha \tag{3.30}$$

où $g(k;\theta)$ est un scalaire. Le vecteur d'état s'écrit de façon similaire comme une somme de deux termes :

$$\begin{pmatrix}
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)} \\
\widehat{\eta}_{(k|k)} \\
\left(\frac{\widehat{\underline{\partial e}}}{\partial t}\right)_{(k|k)}
\end{pmatrix} = \begin{pmatrix}
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)} \\
\widehat{\eta}_{(k|k)} \\
\left(\frac{\widehat{\underline{\partial e}}}{\partial t}\right)_{(k|k)}
\end{pmatrix}_{n,j} + \alpha \underline{f}(k;\theta) \tag{3.31}$$

Comme précédemment, $g(k;\theta)$ et $\underline{f}(k;\theta)$ sont respectivement le produit de $G(k;\theta)$ et $F(k;\theta)$ calculés dans le cas général par la direction du saut a. Nous obtenons :

$$\begin{cases} g(\theta; \theta) = G(\theta; \theta) \ a = H \ a = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0 \\ f(\theta; \theta) = F(\theta; \theta) \ a = K_{(\theta)} \ H \ a = (0) \end{cases}$$

Ce résultat est tout à fait logique puisque, s'il se produit un saut en accélération à l'itération θ , son effet sur la valeur de la vitesse, seule mesure dont nous disposons, ne devient visible qu'à partir de l'itération $\theta + 1$. D'autre part, nous avons :

$$\begin{cases} g(k;\theta) = G(k;\theta) \ a = H \ \Phi^{(k-\theta)} \ a - H \ \Phi \ F(k-1;\theta) \ a \\ \underline{f}(k;\theta) = F(k;\theta) \ a = \Phi \ F(k-1;\theta) \ a + K_{(k)} \ G(k;\theta) \ a \end{cases}$$

$$\begin{cases} g(k;\theta) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \star & \star & (k-\theta) & \Delta t \\ \star & \star & \star & \star \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & \Delta t \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{pmatrix} \underline{f}(k-1;\theta) \\ \underbrace{\underline{f}(k;\theta)}_{\text{et final ement}} = \Phi \underline{f}(k-1;\theta) + K_{(k)} g(k;\theta) \end{cases}$$

et finalement

$$\begin{cases} g(k;\theta) = (k-\theta) \Delta t - \begin{pmatrix} 1 & 1 & \Delta t \end{pmatrix} \underline{f}(k-1;\theta) \\ \underline{f}(k;\theta) = \Phi \underline{f}(k-1;\theta) + K_{(k)} g(k;\theta) \end{cases}$$

Le rapport de vraisemblance défini par l'équation (3.17) s'écrit ici encore sous la forme:

$$l(k;\theta) = \frac{d^2(k;\theta)}{c(k;\theta)}$$

avec
$$c(k;\theta) = \sum_{j=\theta}^k \frac{g^2(j;\theta)}{v_{(j)}}$$
 et $d(k;\theta) = \sum_{j=\theta}^k \frac{g(j;\theta) \; \gamma_{(j)}}{v_{(j)}}$
Si un saut est détecté, l'estimation de son amplitude est de nouveau donnée par :

$$\hat{\alpha}(k; \hat{\theta}) = \frac{d(k; \hat{\theta})}{c(k; \hat{\theta})}$$

L'équation de compensation utilisée dans le filtre de Kalman pour réinitialiser le vecteur d'état s'écrit à présent :

$$\begin{pmatrix}
\begin{pmatrix}
\frac{\widehat{\partial e}}{\partial t}
\end{pmatrix}_{(k|k)} \\
\widehat{\eta}_{(k|k)} \\
\begin{pmatrix}
\frac{\widehat{\partial e}}{\partial t}
\end{pmatrix}_{(k|k)}
\end{pmatrix}_{new} = \begin{pmatrix}
\begin{pmatrix}
\frac{\widehat{\partial e}}{\partial t}
\end{pmatrix}_{(k|k)} \\
\widehat{\eta}_{(k|k)} \\
\begin{pmatrix}
\frac{\widehat{\partial e}}{\partial t}
\end{pmatrix}_{(k|k)}
\end{pmatrix}_{old} + \begin{pmatrix}
(k - \widehat{\theta}) \Delta t \, \widehat{\alpha}(k; \widehat{\theta}) \\
0 \\
\widehat{\alpha}(k; \widehat{\theta})
\end{pmatrix} - \widehat{\alpha}(k; \widehat{\theta}) \underline{f}(k; \widehat{\theta})$$

La compensation se fait ici par ajout à la dernière estimation du vecteur d'état, de l'effet d'un saut en accélération d'amplitude $\hat{\alpha}(k;\theta)$ sur le vecteur d'état pendant $(k-\theta)$ itérations (en particulier, la mise à jour de la vitesse est faite par ajout du produit de la valeur estimée du saut par le temps qui s'est écoulé entre sa production et sa détection), et en retranchant la réponse du filtre de Kalman au saut entre l'instant de sa production et celui de sa détection. Nous procédons aussi à la mise à jour de la matrice de covariance de l'erreur d'estimation par :

$$P_{(k|k)new} = P_{(k|k)old} + [\Phi^{k-\hat{\theta}} - F(k; \hat{\theta})] C^{-1}(k; \hat{\theta}) [\Phi^{k-\hat{\theta}} - F(k; \hat{\theta})]^{T}$$

où $C(k; \hat{\theta})$ est la matrice 3×3 donnée par l'équation générale (3.18).

3.2.4 Filtre de Kalman à accélération constante, saut en vitesse ou en accélération

Dans ce paragraphe, nous cherchons comme dans le cas précédent a détecter des sauts en accélération sous l'hypothèse générale d'un mouvement a accélération constante. Nous cherchons aussi à surveiller d'éventuels sauts en vitesse.

Le calcul du rapport de vraisemblance concernant un saut en accélération est exactement le même que dans le paragraphe précédent.

En ce qui concerne, la détection des sauts en vitesse. La direction du saut dans

l'espace d'état devient $a = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$. Par conséquent, les nouvelles expressions de $g(k;\theta)$ et $f(k;\theta)$ sont :

$$\begin{cases} g(\theta;\theta) = G(\theta;\theta) \ a = H \ a = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = 1 \\ \underline{f}(\theta;\theta) = F(\theta;\theta) \ a = K_{(\theta)} \ H \ a = K_{(\theta)} \end{cases}$$

$$\begin{cases} g(k;\theta) = G(k;\theta) \ a = H \ \Phi^{(k-\theta)} \ a - H \ \Phi \ F(k-1;\theta) \ a \\ \underline{f}(k;\theta) = F(k;\theta) \ a = \Phi \ F(k-1;\theta) \ a + K_{(k)} \ G(k;\theta) \ a \end{cases}$$

$$\begin{cases} g(k;\theta) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & \star & \star \\ \star & \star & \star \\ \star & \star & \star \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & \Delta t \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{pmatrix} \underline{f}(k-1;\theta) \\ \\ \underline{f}(k;\theta) = \Phi \ \underline{f}(k-1;\theta) + K_{(k)} \ g(k;\theta) \\ \\ \underline{f}(k;\theta) = \Phi \ \underline{f}(k-1;\theta) + K_{(k)} \ g(k;\theta) \end{cases}$$

Le rapport de vraisemblance défini par l'équation (3.17) s'écrit ici encore sous la forme:

$$l(k;\theta) = \frac{d^2(k;\theta)}{c(k;\theta)}$$

avec
$$c(k;\theta) = \sum_{j=\theta}^{k} \frac{g^2(j;\theta)}{v_{(j)}}$$
 et $d(k;\theta) = \sum_{j=\theta}^{k} \frac{g(j;\theta) \gamma_{(j)}}{v_{(j)}}$

Des deux rapports de vraisemblance, l'un correpondant à un saut en accélération, l'autre à un saut en vitesse, on retiendra le plus grand pour le comparer a une valeur de seuil fixée ϵ . Si un saut est détecté, l'estimation de son amplitude est de nouveau donnée par :

$$\hat{\alpha}(k; \hat{\theta}) = \frac{d(k; \hat{\theta})}{c(k; \hat{\theta})}$$

Dans le cas particulier d'une détection de saut en vitesse, l'équation de compensation du vecteur d'état s'écrit :

$$\begin{pmatrix}
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)} \\
\widehat{\eta}_{(k|k)} \\
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)}
\end{pmatrix}_{new} = \begin{pmatrix}
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)} \\
\widehat{\eta}_{(k|k)} \\
\left(\frac{\widehat{\partial e}}{\partial t}\right)_{(k|k)}
\end{pmatrix}_{old} + \begin{pmatrix}
\widehat{\alpha}(k; \hat{\theta}) \\
0 \\
0
\end{pmatrix} - \widehat{\alpha}(k; \hat{\theta}) \underline{f}(k; \hat{\theta})$$

Elle est la même que dans le paragraphe précédent en ce qui concerne un saut en accélération. La mise à jour de la matrice de covariance de l'erreur d'estimation est dans les deux cas :

$$P_{(k|k)new} = P_{(k|k)old} + [\Phi^{k-\hat{\theta}} - F(k;\hat{\theta})] C^{-1}(k;\hat{\theta}) [\Phi^{k-\hat{\theta}} - F(k;\hat{\theta})]^T$$

où $C(k; \hat{\theta})$ est la matrice 3×3 donnée par l'équation générale (3.18).

3.3 Résultats expérimentaux

Pour les deux tâches précédemment décrites dans le paragraphe 2.4, le suivi a été réalisé en comparant les quatre méthodes suivantes :

- sans estimer le mouvement de l'objet. Nous devons alors observer des erreurs de traînage pendant le suivi.
- en estimant le mouvement de l'objet avec un filtre de Kalman à vitesse constante et bruit corrélé (sans GLR).
- en estimant le mouvement de l'objet avec un filtre de Kalman à accélération constante et bruit corrélé. Ce filtre doit réagir plus rapidement que le filtre précédent en cas de changement brusque en vitesse. Il doit cependant être également plus sensible au bruit.
- en utilisant le test du GLR pour détecter des sauts, soit en vitesse, soit en accélération.

3.3.1 Suivi d'un carré par liaison rigide

Cette première tâche de suivi d'un carré par liaison rigide contraint les six degrés de liberté de la caméra. Elle consiste à garder la position et l'orientation de la caméra par rapport au carré constantes.

L'expérimentation est réalisée comme suit : le carré étant immobile, nous commençons par ramener par asservissement visuel la caméra de sa position initiale (voir Figure 2.4.a) à la position désirée (voir Figure 2.4.b). Le carré débute ensuite un mouvement de translation avec une vitesse constante de 5 cm/s sur une longueur de 90 cm. Après un arrêt, le carré retourne à sa position de départ avec une vitesse de 10 cm/s.

3.3.1.1 Résultats sans estimation du mouvement de l'objet

Sur la Figure 3.2 sont représentés les résultats obtenus quand l'estimation du mouvement de l'objet n'est pas effectuée (i.e., $\frac{\widehat{\partial \underline{e}}}{\partial t} = 0$ dans (2.6)). Plus précisément, la Figure 3.2 a montre à chaque itération de la loi de commande la valeur des quatre premières composantes de $(\underline{s} - \underline{s}^*)$ (c'est-à-dire les coordonnées en X de l'erreur observée dans l'image) et la Figure 3.2 b les quatre dernières composantes de $(\underline{s} - \underline{s}^*)$ (c'est-à-dire les coordonnées en Y de l'erreur observée dans l'image). Finalement, la Figure 3.2 c donne les valeurs successives de la vitesse de la caméra calculées par la loi de commande. Sur cette figure ne sont représentées que les trois composantes de la vitesse de translation, car le carré n'est animé que d'un mouvement de translation (la vitesse de rotation de la caméra est donc toujours quasiment nulle).

Quand l'objet est immobile, nous observons une erreur maximale de 1 pixel dans l'image à laquelle correspond une commande de moyenne nulle et d'écart type 0.05 cm/s. Ces très légères erreurs sont dûes aux erreurs de mesure effectuées pour déterminer la position dans l'image des quatre points considérés.

Lorsque le carré se déplace à vitesse constante (des itérations 320 à 770 et 1200 à 1425), l'erreur observée dans l'image dûe à ce mouvement entraîne un mouvement de poursuite de la caméra. Cependant, la loi de commande ne permet pas de compenser entièrement ces erreurs, car l'absence d'estimation du mouvement de l'objet dans l'image induit une erreur de traînage importante (et constante puisque l'objet se déplace à vitesse constante) d'environ 110 et 48 pixels respectivement sur les axes horizontal et vertical de l'image, correspondant à une vitesse de 5 cm/s du carré. L'amplitude de cette erreur de traînage dépend bien évidemment, de la valeur du gain λ et de l'amplitude de la vitesse de l'objet, de sorte que lorsque le carré se déplace à une vitesse de 10 cm/s, l'erreur de traînage est tellement grande qu'elle entraîne la disparition de l'objet du champ de vision de la caméra et l'arrêt du suivi.

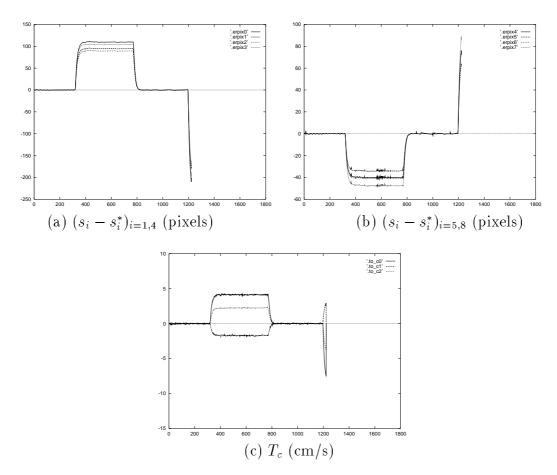


Figure 3.2 : Suivi d'un carré sans estimation de son mouvement

3.3.1.2 Filtre de Kalman à vitesse constante et bruit corrélé

Les résultats obtenus pour le suivi du carré en introduisant dans la commande l'estimation du mouvement de l'objet calculée à l'aide d'un filtre de Kalman à vitesse constante et bruit corrélé sont donnés sur la Figure 3.3. Lorsque l'objet est à l'arrêt, on peut observer les mêmes erreurs que dans le cas précédent, ce qui indique que la vitesse nulle de l'objet est correctement estimée, et que le filtrage de Kalman ne perturbe pas la loi de commande. Lorsque le carré est en mouvement stationnaire, les erreurs de traînage sont à présent supprimées puisque l'on obtient une erreur maximale de 1.5 pixels dans l'image. La vitesse du carré est alors correctement estimée avec un écart type de 0.06 cm/s (voir Figure 3.3.d). Nous pouvons également remarquer le bon lissage de l'estimation par comparaison avec la valeur mesurée (voir Figure 3.3.c).

Au début et à la fin du mouvement du carré, le filtre a besoin d'une centaine d'itérations pour estimer correctement la nouvelle vitesse de l'objet dans l'image. Plus précisément, 120 itérations sont nécessaires pour obtenir une erreur dans l'image inférieure à 4 pixels. Ceci a pour effet une erreur maximale d'environ 65 (respectivement 137) et 27 (respectivement 55) pixels sur l'axe horizontal et respectivement vertical de l'image pour une vitesse du carré de 5 cm/s (respectivement 10 cm/s). Concernant la vitesse de la caméra (voir Figure 3.3.e), nous pouvons observer un dépassement au début du mouvement du carré (qui permet de supprimer les erreurs de traînage) et une oscillation à la fin de ce mouvement (dont la durée dépend du temps nécessaire à la convergence du filtre).

Précisons en ce qui concerne la mise au point du filtre de Kalman, que la valeur de la variance sur l'erreur de mesure σ_R^2 a été estimée à la suite de mesures effectuées en appliquant une commande nulle au robot. La variance de la variable d'état est elle fixée empiriquement pour obtenir un filtrage satisfaisant.

3.3.1.3 Filtre de Kalman à accélération constante et bruit corrélé

Les résultats obtenus en adoptant ce modèle sont donnés sur la Figure 3.3. En régime permanent, nous observons, dans ce cas d'utilisation d'un filtre de Kalman à accélération constante et bruit corrélé, une erreur maximale de 4 pixels sur les axes horizontal et vertical de l'image (voir Figure 3.4). La vitesse du carré est estimée avec un écart type de 0.2 cm/s et la consigne de commande appliquée au robot présente un écart type de 0.3 cm/s. Ces valeurs plus importantes que dans les cas précédents indiquent que ce filtre est peu robuste aux bruits de mesure, interprétés comme possible accélération. Il en résulte une estimation imprécise du mouvement du carré, d'où une erreur et une consigne de commande plus importantes et moins

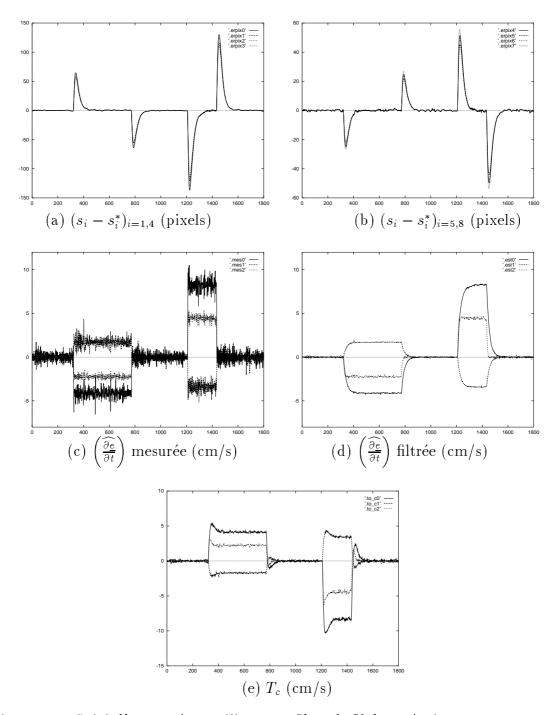


Figure 3.3 : Suivi d'un carré en utilisant un filtre de Kalman à vitesse constante et bruit corrélé

stables que précédemment.

La réponse au brusque changement d'amplitude du mouvement de l'objet est cependant beaucoup plus rapide, permettant de réduire significativement l'erreur observée dans l'image. En effet, nous observons lors du changement de vitesse une erreur maximale d'environ 48 (92) et 18 (38) pixels sur l'axe horizontal et vertical de l'image pour une vitesse du carré de 5 cm/s (10 cm/s), et seulement 40 itérations sont nécessaires pour avoir une erreur dans l'image équivalente à celle enregistrée quand le carré est à l'arrêt.

3.3.1.4 Test du GLR sur un filtre de Kalman à vitesse constante et bruit corrélé

Les résultats obtenus dans ce dernier cas où l'algorithme du GLR est utilisé pour détecter et compenser des changements brusques de la vitesse de la cible sont représentés sur la Figure 3.5. La Figure 3.5.f représente les instants et les amplitudes des ruptures de modèles détectés à l'aide de l'agorithme du GLR dans le cas d'un modèle à vitesse constante et bruit corrélé (voir paragraphe 4.2).

A l'arrêt ou lorsque la vitesse du carré est constante, nous obtenons des résultats équivalents à ceux obtenus avec un filtre de Kalman à vitesse constante et bruit corrélé. Ces résultats sont donc plus stables et robustes que dans le cas précédent du filtre à accélération constante et bruit corrélé. Ce résultat est logique vu l'absence de fausses détections de rupture de modèle au cours du suivi de l'objet. Cette absence de fausses détections est consécutive à notre bon réglage de la valeur du seuil de détection des sauts ϵ (fixé à 5).

Au début et à la fin du mouvement de translation de l'objet, nous pouvons observer une erreur maximale d'environ 45 (87) et 25 (35) pixels sur l'axe horizontal et vertical de l'image pour une vitesse du carré de 5 cm/s (10 cm/s). Les sauts en vitesse de l'objet sont correctement détectés et estimés par le test du GLR. De plus, la réponse à ces sauts est satisfaisante puisque nous obtenons de nouveau une erreur dans l'image inférieure à 4 pixels après 40 itérations. Lors des ruptures de modèle du mouvement de l'objet, les performances obtenues ici sont donc meilleures à celles du filtre de Kalman à accélération constante et bruit corrélé.

L'utilisation du test du GLR pour estimer et compenser la vitesse de l'objet est donc optimale dans cet exemple de poursuite puisqu'elle permet de conjuguer les qualités des deux solutions précédentes (stabilité et robustesse d'une part, bonne estimation et compensation des ruptures d'autre part). Signalons que nous avons

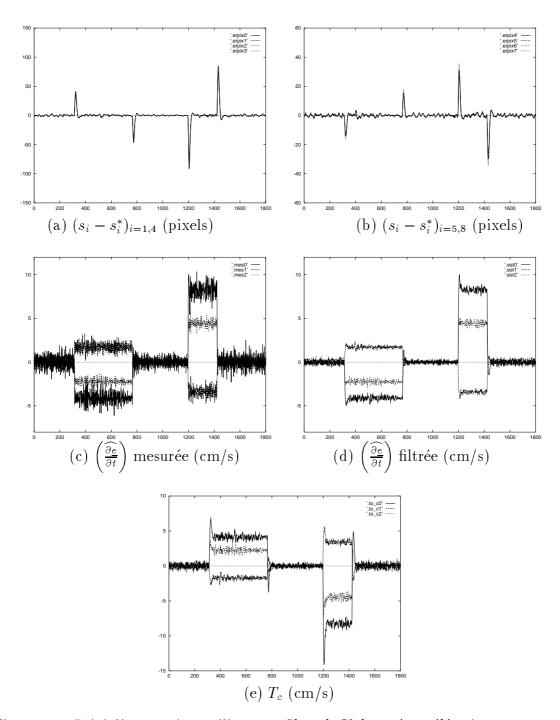


Figure 3.4 : Suivi d'un carré en utilisant un filtre de Kalman à accélération constante et bruit corrélé

utilisé une fenêtre de détection de taille M=10, car la détection des sauts est effectuée en moyenne avec seulement une itération de retard par rapport à l'instant de sa production.

3.3.2 Suivi d'un point en contrôlant l'orientation de la caméra

Pour cette seconde tâche de suivi d'un objet mobile, la caméra est commandée afin que l'objet poursuivi apparaisse toujours au centre de l'image.

L'expérimentation est réalisée de la façon suivante : nous débutons par une phase d'initialisation qui consiste à ramener l'image du point considéré au centre de l'image en utilisant la loi de commande (2.11) (voir Figure 2.5). Puis, le robot effectue un mouvement secondaire afin de simuler un mouvement opposé de la cible. Ce mouvement (voir Figure 3.6.a) consiste en diverses translations parallèles au plan image ($V_x = 8 \text{ cm/s}, V_y = 4 \text{ cm/s}$ des itérations 100 à 300 et 800 à 1000, $V_x = -8 \text{ cm/s}, V_y = -4 \text{ cm/s}$ des itérations 450 à 650 et 1150 à 1350), qui doivent être compensées par des rotations de la caméra afin que l'objet apparaisse toujours au centre de l'image.

3.3.2.1 Résultats sans estimation du mouvement de l'objet

Les résultats correspondant au cas où l'estimation de la vitesse de l'objet n'est pas effectuée sont donnés sur la Figure 3.6. Nous obtenons les mêmes phénomènes que ceux observés lors du suivi du carré (voir paragraphe 5.1.1) : au repos, nous observons une erreur maximale de 1/2 pixel (voir Figure 3.6.b). Par contre, comme nous pouvions le prévoir, la non-estimation du mouvement de l'objet pendant la phase de translation entraîne une erreur permanente de 10 et 5 pixels respectivement sur les axes horizontal et vertical de l'image (voir Figure 3.6.b).

3.3.2.2 Filtre de Kalman à vitesse constante et bruit corrélé

En utilisant un filtre de Kalman à vitesse constante et bruit corrélé pour estimer le mouvement de l'objet, nous observons quand ce mouvement est nul une erreur d'écart type 1/4 pixel sur les axes horizontal et vertical de l'image (voir Figure 3.7.b). Les estimées des vitesses du pan et du tilt sont nulles et de même écart type 0.02 dg/s (voir Figure 3.7.d). Nous remarquons le bon filtrage du bruit par comparaison avec la Figure 3.7.c où est représentée la mesure de $\frac{\widehat{\partial \varepsilon}}{\partial t}$. Les consignes de commande correspondantes sont stables puisque de moyenne nulle et d'écart type 0.07 dg/s (voir Figure 3.7.e).

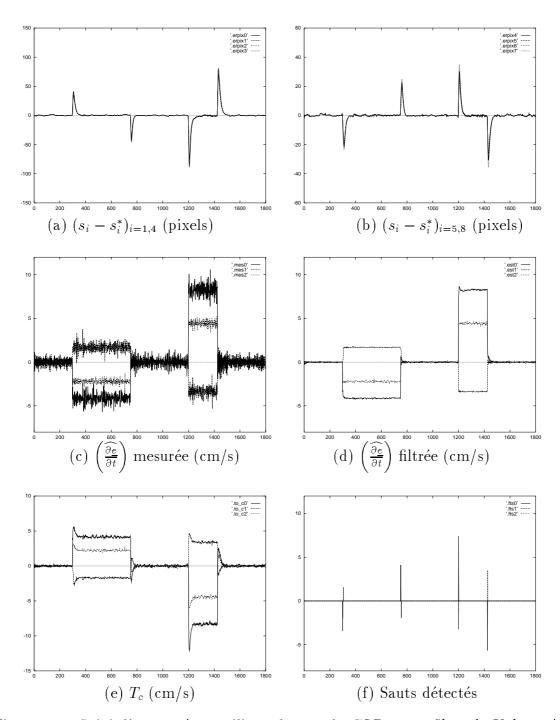


Figure 3.5 : Suivi d'un carré en utilisant le test du GLR et un filtre de Kalman à vitesse constante et bruit corrélé

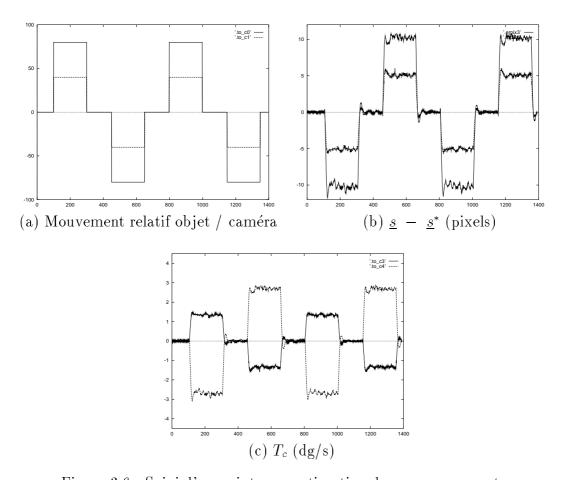


Figure 3.6 : Suivi d'un point sans estimation de son mouvement

Au début et à la fin d'un mouvement de translation autour de l'objet, le filtre de Kalman nécessite environ 60 itérations pour estimer correctement la nouvelle vitesse de l'objet. Ceci provoque des erreurs maximales de 10 et 5 pixels respectivement sur les axes horizontal et vertical de l'image. Ce faible temps de réponse peut être amélioré par l'utilisation des méthodes de filtrage décrites ci-dessous.

3.3.2.3 Filtre de Kalman à accélération constante et bruit corrélé

Les résultats obtenus en utilisant un filtre de Kalman à accélération constante et bruit corrélé sont donnés sur la Figure 3.8, configurée comme la Figure 3.7. Lorsque l'objet est immobile, nous observons une erreur d'écart type 1/4 pixel sur les axes horizontal et vertical de l'image. L'estimée de la vitesse du pan et du tilt sont nulles d'écart type 0.07 dg/s. Les consignes de commande correspondantes sont de moyenne nulle et d'écart type 0.14 dg/s. Bien que nous observons une erreur dans l'image équivalente à celle obtenue avec un filtre de Kalman à vitesse constante et bruit corrélé, nous remarquons que l'estimation du mouvement du point est beaucoup plus bruitée, ce qui entraîne des consignes de commande moins stables et d'amplitudes plus importantes.

Au début et à la fin du mouvement de translation, nous obtenons des erreurs maximales d'environ 8 et 5 pixels sur les axes horizontal et vertical de l'image. Ces résultats sont meilleurs que ceux obtenus par la méthode décrite précédemment, puisque cette modélisation du filtre de Kalman permet d'estimer la nouvelle vitesse du point plus rapidement : en effet, une erreur dans l'image inférieure à 1 pixel est obtenue après seulement 40 itérations de la boucle de commande.

3.3.2.4 Test du GLR et filtre de Kalman à vitesse constante et bruit corrélé

Ce cas où nous avons utilisé le test du GLR pour détecter les sauts en vitesse de l'objet permet d'obtenir les meilleurs résultats (voir Figure 3.9).

Au repos, nous observons une erreur d'écart type 1/4 pixel sur les axes horizontal et vertical de l'image. Les estimées des vitesses du pan et du tilt sont nulles de même écart type de 0.02 dg/s. Les consignes de commande correspondantes sont de moyenne nulle et d'écart type 0.07 dg/s. Nous remarquons que ces performances sont exactement les mêmes que celles obtenues avec un filtre de Kalman à vitesse constante, le test du GLR n'ayant détecté aucune rupture de modèle.

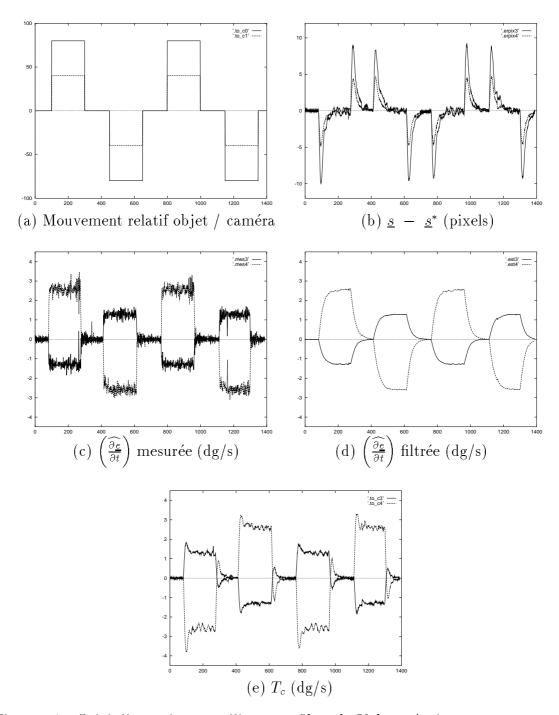


Figure 3.7 : Suivi d'un point en utilisant un filtre de Kalman à vitesse constante et bruit corrélé

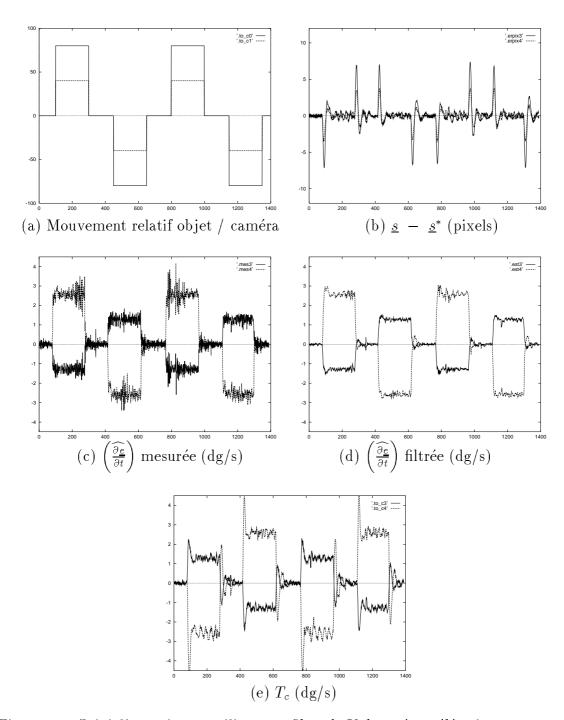


Figure 3.8 : Suivi d'un point en utilisant un filtre de Kalman à accélération constante et bruit corrélé

Au début et à la fin des mouvements de translation autour du point cible, les sauts en vitesse sont correctement détectés et estimés (voir Figure 3.9.f) et nous observons de nouveau des erreurs maximales d'environ 8 et 5 pixels sur les axes horizontal et vertical de l'image. De plus, 25 itérations sont nécessaires pour avoir une erreur dans l'image inférieure à 1 pixel. Le test du GLR permet donc d'obtenir un temps de réponse plus court que le filtre de Kalman à accélération constante et bruit corrélé, sans en avoir les inconvénients.

Comme dans le cas précédent du suivi du carré, la taille de la fenêtre de détection a été fixée à M=10 car les sauts sont ici détectés en environ deux itérations.

3.3.3 Suivi d'un point animé d'un mouvement accéléré

Nous utilisons ici encore les deux degrés de liberté en rotation de la caméra (pan Ω_x et tilt Ω_y), afin que l'image de l'objet apparaisse en permanence au centre de l'image. Le mouvement secondaire du robot (qui permet rappelons-le de simuler un mouvement opposé de l'objet) consiste en une translation le long de l'axe x du repère caméra avec une succession d'accélérations et de décélérations (voir Figure 3.10.a).

3.3.3.1 Résultats sans estimation du mouvement de l'objet

La non estimation du mouvement de l'objet entraîne l'apparition d'une erreur de traînage qui suit le modèle du mouvement de l'objet et qui peut atteindre 30 pixels sur l'axe horizontal de l'image (voir Figure 3.10.b). Cet axe est le seul auquel nous nous intéressons dans notre comparatif, puisque l'erreur est toujours nulle sur l'axe vertical de l'image (au bruit près). En effet, le mouvement relatif de l'objet par rapport à la caméra se situe uniquement dans le plan défini par les axes x et z du repère caméra.

3.3.3.2 Filtre de Kalman à vitesse constante et bruit corrélé

Ici, le modèle de mouvement considéré dans le filtre de Kalman est d'un ordre inférieur au mouvement réel, raison pour laquelle nous observons un retard de 70 itérations dans l'estimation de la vitesse de l'objet dans l'image (voir Figure 3.11.c). Ce retard provoque une erreur permanente dans l'image dont la valeur maximale est d'environ 10 pixels (voir Figure 3.11.b).

3.3.3.3 Filtre de Kalman à accélération constante et bruit corrélé

Dans ce cas, le modèle de mouvement retenu correspond au mouvement réel, ce qui entraîne la disparition du retard d'estimation en régime stationnaire (voir Fi-

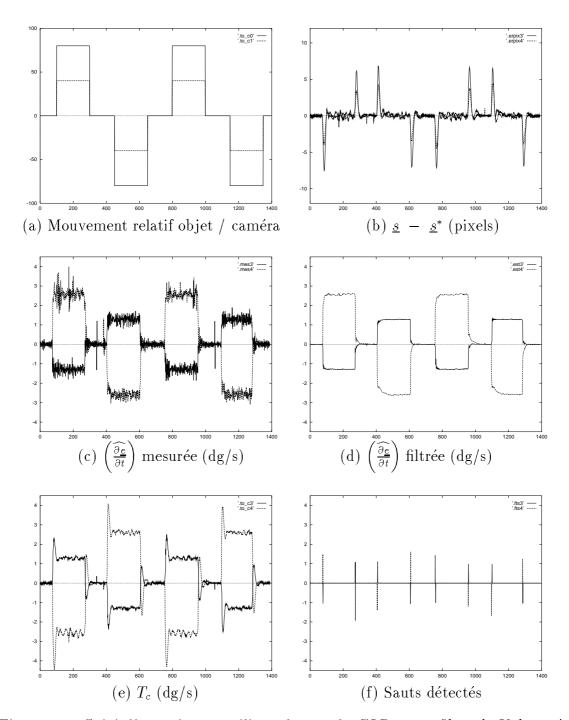


Figure 3.9 : Suivi d'un point en utilisant le test du GLR et un filtre de Kalman à vitesse constante et bruit corrélé

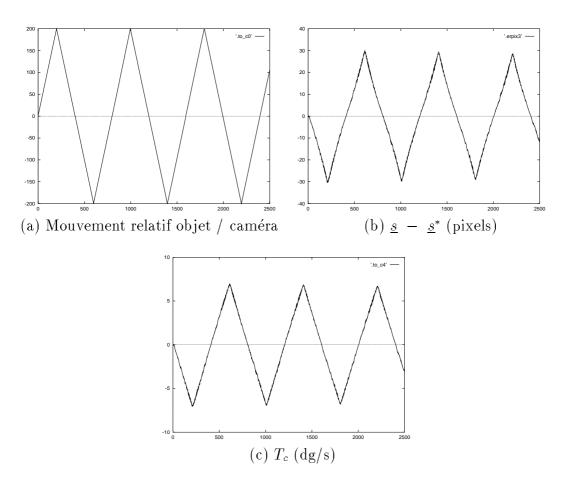


Figure 3.10 : Suivi d'un point animé d'un mouvement accéléré sans estimation de son mouvement

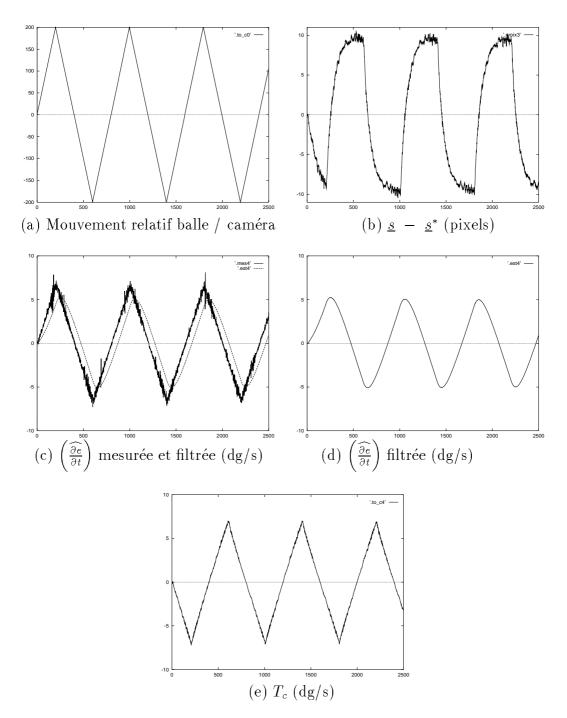


Figure 3.11 : Suivi d'un point animé d'un mouvement accéléré en utilisant un filtre de Kalman à vitesse constante et bruit corrélé

gure 3.12.c). Par contre, on obtient une mauvaise estimation de la vitesse de l'objet lors des changements en accélération (voir Figure 3.12.c) et un temps de réponse important pour estimer correctement la nouvelle valeur de l'accélération (voir Figure 3.12.d). L'erreur maximale dans l'image est maintenant égale à 5.5 pixels (voir Figure 3.12.b). Notons qu'il faut 150 itérations pour obtenir une erreur stationnaire de 1 pixel lors des changements d'accélération.

3.3.3.4 Détection de sauts en accélération avec le test du GLR sur un filtre de Kalman à accélération constante et bruit corrélé

Dans ce dernier cas, les changements d'accélération sont bien détectés (voir Figure 3.13.f), ce qui entraîne la suppression du retard d'estimation de la nouvelle valeur de l'accélération (voir Figure 3.13.d). Nous pouvons observer la mise à jour de la valeur de la vitesse après chaque détection de saut en accélération (voir Figure 3.13.d) et son effet sur la vitesse de la caméra (voir Figure 3.13.e). Nous obtenons une erreur maximale de 4.5 pixels (voir Figure 3.13.b) et il faut seulement 30 itérations pour rétablir une erreur dans l'image de 1 pixel après un saut en accélération.

La fenêtre de détection utilisée ici est de taille M=50, car la détection du saut s'effectue avec un retard de 25 itérations en moyenne. Ce retard, plus important que dans les cas précédents de détection de saut en vitesse, est dû au fait que l'on détecte un saut dans la dérivée, non mesurée, d'un signal bruité.

Dans le cas des expérimentations que nous venons de décrire, le modèle de mouvement retenu correspondait bien au modèle réel. Dans le paragraphe suivant, nous allons étudier les performances dans le cas d'absence de cette correspondance.

3.3.4 Suivi d'un point animé d'un mouvement complexe

Le but de cette expérimentation est semblable au précédent exemple. Il s'agit de la même manière de commander le pan et le tilt de la caméra pour maintenir le centre de gravité de l'objet poursuivi au centre de l'image.

La cible est animé ici d'un mouvement complexe, combinaison d'une rotation uniforme et de translations en mouvement accéléré. Plus précisément, le robot se déplace avec une succession d'accélérations et de décélérations au niveau de ses trois degrés de liberté en translation, et la cible fixée sur un tourne disque décrit un mouvement circulaire uniforme. Il en résulte un mouvement complexe de l'objet dans l'image (see Figure 3.15.b).

En comparant la figure 3.14.a et la figure 3.15.a, nous remarquons que la prise en considération du mouvement de la cible permet de réduire de façon sensible l'erreur

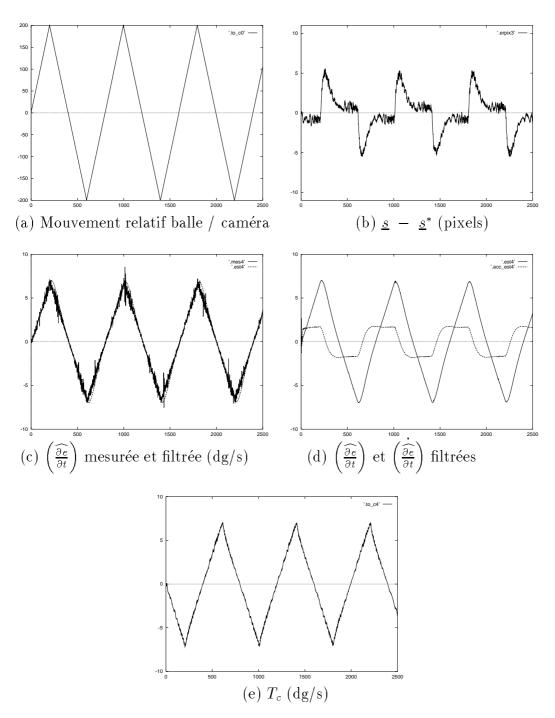


Figure 3.12 : Suivi d'un point animé d'un mouvement accéléré en utilisant un filtre de Kalman à accélération constante et bruit corrélé

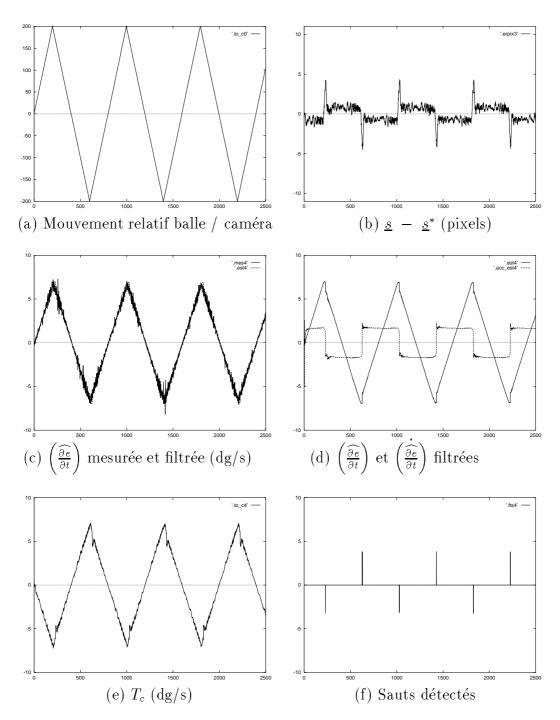


Figure 3.13 : Suivi d'un point animé d'un mouvement accéléré en utilisant le test du GLR et un filtre de Kalman à accélération constante et bruit corrélé

Conclusion 81

de poursuite dans l'image malgré la mauvaise estimation de la vitesse de la cible obtenue avec un filtre de Kalman a vitesse constante.

L'hypothèse que la cible possède un mouvement à accélération constante permet un meilleur suivi. Nous obtenons des erreurs dans l'image plus faibles que dans le cas où nous utilisons un filtre de Kalman a vitesse constante (voir Figure 3.16.a). Ce résultat découle du fait que les estimées de la vitesse de la cible sont plus exactes.

L'utilisation du test du GLR en parallèle à un filtre de Kalman à accélération constante, réduit de façon significative l'erreur dans l'image (voir Figure 3.17.a). Nous remarquons quelques sauts au niveau de l'estimation de la vitesse de la cible (voir Figure 3.17.c) correspondants aux opérations de mise à jour quand des sauts en accélération sont détectés.

Précisons toutefois qu'un réglage assez fin des variances de bruit du filtre de Kalman (acceptant de fortes variations de l'état du filtre) permet d'atténuer les différences de performances entre les cas d'utilisation ou non du test du GLR.

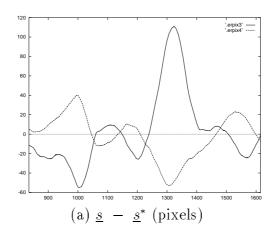


Figure 3.14 : Suivi d'une cible animée d'un mouvement complexe sans estimation de son mouvement

3.4 Conclusion

Nous avons présenté dans ce chapitre un schéma d'asservissement visuel utilisant l'approche fonction de tâche. Ce schéma spécifie la tâche de vision en termes de régulation dans l'image. Nous avons proposé une loi de commande adaptative et prédictive basée sur cette approche qui permet de poursuivre un objet possédant un mouvement inconnu. Pour ce faire, un schéma robuste d'estimation et de prédiction du mouvement de la cible dans l'image a été présenté et introduit dans la loi de

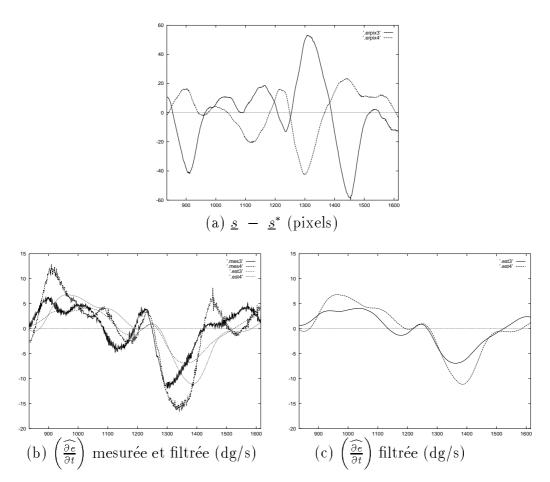


Figure 3.15 : Suivi d'une cible animée d'un mouvement complexe en utilisant un filtre de Kalman à vitesse constante et bruit corrélé

Conclusion 83

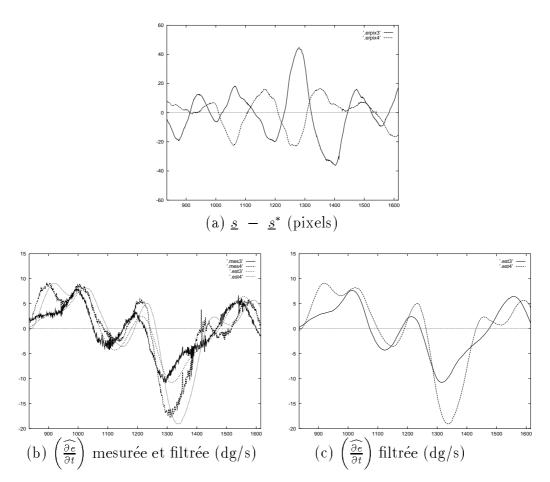


Figure 3.16 : Suivi d'une cible animée d'un mouvement complexe en utilisant un filtre de Kalman à accélération constante et bruit corrélé

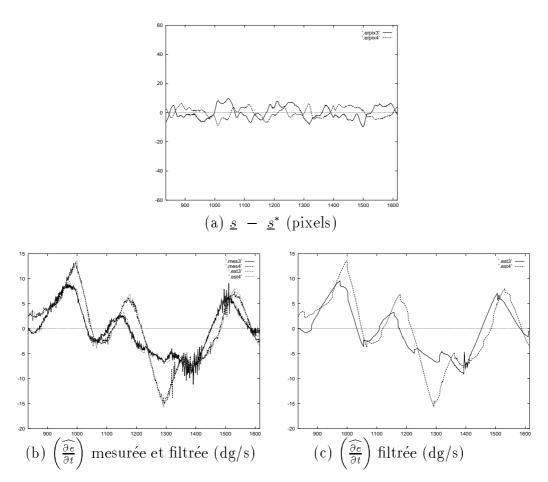


Figure 3.17 : Suivi d'une cible animée d'un mouvement complexe en utilisant le test du GLR et un filtre de Kalman à accélération constante et bruit corrélé

Conclusion 85

commande. Les résultats expérimentaux soulignent le fait que l'emploi d'un filtre de Kalman combiné avec un test du GLR pour estimer le mouvement de la cible permet de détecter et de compenser des changements brusques du mouvement de la cible.

Il ressort des expérimentations réalisées l'importance du modèle retenu du filtre de Kalman. Par conséquent, les résultats se trouvent améliorés si on a un bon apriori sur le type de mouvement de l'objet poursuivi. Le GLR se prête bien à des tâches particulières d'asservissement visuel, car il est très bien adapté à des mouvements particuliers, comme par exemple, des opérations de manipulation au niveau de chaînes de montage. En effet, ces dernières se caractérisent par des mouvements saccadés.

Chapitre 4

Estimation du mouvement 3D d'un objet par vision active

Dans ce chapitre, nous développons un schéma pour l'estimation du mouvement 3D d'un objet à partir d'images 2D fournies par une caméra montée sur l'effecteur d'un robot. Après un rapide état de l'art, nous nous focalisons sur l'estimation du mouvement 3D d'un point. Ce cas est intéressant car l'ensemble des composantes de son mouvement n'est alors pas directement observable. Pour remonter au mouvement spatial de l'objet, deux points de vue distincts sont nécessaires. Dans une première étape, l'estimation des composantes de la vitesse 3D d'un point est faite en utilisant les informations fournies par la caméra à partir de deux positions fixes. Dans la seconde partie, nous cherchons à résoudre le même problème en exploitant les données images correspondant à deux positions distinctes de la caméra alors que celle-ci est en mouvement. Pour les deux cas, nous avons considéré l'hypothèse selon laquelle le point possède un mouvement rectiligne uniforme. Cette hypothèse permet de considérer qu'il n'y a pas de changement dans les paramètres du mouvement de l'objet entre les deux instants correspondants aux deux positions de la caméra considérées. Précisons que la résolution du système d'équations obtenues nous permet de remonter aussi à la profondeur du point pour chacune des deux positions. Nous obtenons donc la vitesse et la position du point dans l'espace 3D.

4.1 Etat de l'art

Nous ne décrirons pas ici les multiples travaux effectués dans le domaine utilisant un capteur stéréoscopique (fixe, mobile ou actif) mais nous nous focaliserons au cas où une seule caméra est utilisée.

De nombreux travaux de recherche se sont intéressés au problème de reconstruction 3D d'une scène fixe en absence d'informations fiables sur le déplacement de la caméra. Dans ce cas, les méthodes de résolution développées aboutissent à la détermination de la structure 3D de la scène perçue et du mouvement de la caméra. Remarquons que le problème qui nous intéresse est à peu près équivalent. En effet, dans le cas général, le mouvement relatif caméra scène est inconnu. Dans notre cas, le mouvement de la scène (réduite à un point) est inconnu alors que le mouvement de la caméra lui est parfaitement connu. Nous verrons que le système d'équations obtenu nous fournit la connaissance du mouvement 3D du point ainsi que sa profondeur par rapport à la caméra, ce qui revient à la connaissance de sa position spatiale, étant donné que la position de la caméra est disponible.

Nous avons également été confrontés (mais dans un cas particulier seulement) au résultat classique de la reconstruction à partir du mouvement qui concerne l'ambiguïté échelle/vitesse (le mouvement rapide d'un observateur éloigné de la scène est indiscernable du mouvement lent d'un observateur proche de la scène). Autrement dit, on peut multiplier par le même facteur proportionnel la valeur de la profondeur du point et l'amplitude de la vitesse de translation de la caméra tout en observant aucune différence dans l'image. En effet, nous montrons que si la caméra effectue un seul mouvement de translation, alors nous retrouvons l'ambiguïté du mouvement tridimensionnel du point. Par contre, nous verrons que si la caméra est également animée d'un mouvement de rotation, alors cette ambiguïté disparaît.

Parmi les recherches qui ont été menées dans ce domaine de la détermination de la structure de la scène et du mouvement de la caméra, la plupart utilise comme observations introduites dans la formulation du problème des points et leurs projections [Faugeras et al.87], [Weng et al.89], [Boukarri89], [Wu et al.88], [Thompson et al.93] ou des droites 3D et leurs projections [Liu et al.86], [Faugeras et al.87], [Weng et al.92], [Mitiche et al.89]. Dans ce dernier cas, deux images sont insuffisantes pour déterminer le mouvement de la caméra. Un minimum de trois images est en effet requis. Cela provient du fait que les droites 3D possèdent un degré de liberté supplémentaire par rapport aux points 3D.

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

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

• les algorithmes non linéaires qui exploitent directement les équations non li-

Etat de l'art 89

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

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

Nous présentons ci-dessous une liste non exhaustive de travaux qui ont été développés dans les deux directions. Le lecteur intéressé pourra se reférer à la synthèse bibliographique (peu récente certes) réalisée à ce sujet par Aggarwal [Aggarwal et al.88].

Weng et al. dans [Weng et al.89] proposent une méthode de calcul du mouvement et de la structure 3D d'une scène à partir de la mise en correspondance de points sur deux images. L'algorithme requiert un minimum de huit points mais il en exploite un nombre plus important afin d'améliorer la robustesse au bruit. La structure 3D des points est alors estimée par moindres carrés.

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

- le champ de vue de la caméra doit être large pour obtenir de meilleures estimées.
- l'amplitude de la translation doit être importante pour permettre une estimation stable de la direction de la translation et de la structure (ce résultat est classique et se retrouve pour les systèmes stéréoscopiques).

- une translation parallèle au plan image assure une estimation plus stable.
- la scène doit être proche de la caméra.
- les paramètres de rotation n'ont pas d'influence significative sur la fiabilité des solutions.

De plus, dans [Faugeras et al.90] est donné le nombre de solutions du mouvement possible entre deux images à partir de plusieurs points mis en correspondance.

Liu et Huang, [Liu et al.86], ont développé un algorithme pour l'estimation du mouvement de la caméra à partir de la mise en correspondance de droites 3D d'une scène rigide. Les principes de base sont les mêmes que les algorithmes basés sur les points 3D.

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

Une approche plus robuste qui intègre de manière explicite les incertitudes sur les mesures est celle présentée dans [Faugeras et al.87]. Cette méthode utilise un filtre de Kalman étendu afin d'obtenir une meilleure estimation des paramètres du mouvement de la caméra. La structure des droites 3D est alors déterminée en utilisant soit une technique de moindres carrés, soit un filtre de Kalman.

Dans [Weng et al.92], les auteurs proposent aussi une méthode linéaire pour l'estimation du mouvement et de la structure 3D basée sur l'utilisation de droites. Trois images sont utilisées et un minimum de treize droites est requis dans cette formulation. Cette approche garantit une solution unique pour les paramètres de mouvement en dehors de certains cas dégénérés. Les auteurs présentent quelques conditions nécessaires et suffisantes sur les configurations de droites conduisant à une matrice de coefficients dégénérée. En présence de bruit, cette méthode linéaire n'est pas suffisamment robuste. Ils proposent donc une optimisation non linéaire initialisée par leur algorithme linéaire. Cette approche consiste à minimiser les erreurs observées dans les images par rétro-projection. Plus précisément, on minimise l'écart entre les valeurs estimées des paramètres des droites (calculés par moindres

Etat de l'art

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

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

Enfin, notons qu'il existe des approches qui combinent les deux types de primitives (points et droites). Dans ce cas, quatre points et une droite sur deux images sont suffisantes pour calculer la structure et le mouvement 3D [Aggarwal et al.87]. Le cas d'un système non calibré a été récemment étudié dans [Viéville et al.96a], fournissant une estimation de la structure à une transformation projective près.

On peut également citer la méthode originale de factorisation dûe à Tomasi et Kanade [Tomasi et al.92]. Cette méthode valide pour un modèle de projection orthographique, permet d'isoler d'une part, la structure 3D de la scène, et d'autre part, le mouvement de la caméra. Cette méthode a été récemment généralisée à un modèle de projection perspective [Christy et al.96].

Assez peu de travaux se sont intéressés au cas d'une caméra fixe et d'objets en mouvement. On peut citer les travaux (déjà anciens) de Tsai et Huang ([Tsai et al.81] pour le cas d'un objet plan et [Tsai et al.84] pour une généralisation à des objets quelconques). Le principe est basé sur l'appariement dans les deux images considérées de plusieurs points appartenant à l'objet en mouvement. La prise en compte de séquences d'images plus longues a été effectuée par Broida et Chellapa ([Broida et al.86] et [Broida et al.91]) où sont enfin présentés des résultats expérimentaux). L'utilisation d'un filtre de Kalman étendu permet de robustifier la solution trouvée en présence de bruit.

Enfin dans le cas où la caméra est en mouvement et un (ou plusieurs) objets sont mobiles dans la scène, on retrouve les travaux de Adiv [Adiv85]. L'approche consiste en deux étapes fondamentales : dans la première phase, le mouvement apparent est segmenté en régions élémentaires ; chaque région étant cohérente avec le mouvement rigide d'une surface plane. L'ensemble des vitesses apparentes appartenant à une même région est alors utilisé pour calculer les paramètres de mouvement

correspondants, optimaux au sens des moindres carrés. La seconde phase consiste à regrouper les régions dont les mouvements 3D correspondants sont proches. On suppose encore, en s'appuyant sur la contrainte de rigidité, que de tels groupes de régions correspondent au mouvement d'un même objet rigide. Une fois que le champ des vitesses apparentes a été segmenté, le but est de déterminer les paramètres du mouvement 3D relatif caméra-objet et les profondeurs relatives des points correspondants dans chaque région. Pour cela, une technique de moindres carrés est utilisée. Elle consiste à minimiser l'écart entre les valeurs mesurées du mouvement 2D et celles prédites à partir de la structure et du mouvement 3D estimés.

Les études menées dans [Cipolla et al.92], [Subbarao89] et [Meyer93] se situent dans la suite de ces travaux. L'approche consiste à utiliser les composantes du mouvements apparent ou les paramètres affines de mouvement 2D, obtenus à l'issue d'une segmentation en régions, pour remonter à une information sur le mouvement 3D des objets (comme par exemple le temps à collision) et leur structure 3D (orientation de la surface).

Finalement, une approche d'estimation du mouvement 3D par vision active est décrite dans [Bandopadhay et al.86], [Aloimonos et al.87] et [Aloimonos et al.91]. Plus exactement, le but de ces travaux est similaire à ceux décrits au début de ce chapitre et consiste à recouvrer le mouvement 3D d'un observateur se déplaçant dans un environnement statique. Ils montrent alors qu'un observateur actif résoud le problème de la structure à partir du mouvement de manière plus efficace en utilisant une ou deux caméras. Dans le cas qui nous concerne, un observateur monoculaire actif ramène la dimension de l'espace des inconnues (paramètres du mouvement de l'observateur) de 5 à 4, sans toutefois s'affranchir des non linéarités. Si l'observateur monoculaire peut discerner et effectuer la poursuite d'une caractéristique pertinente de la surface observée (un point ici), le problème est alors simplifié. En supposant que la vitesse apparente du point caractéristique choisi est connue avec précision, alors la dimension des inconnues des équations du champ 2D de vitesses apparentes est réduite de 1 sans augmenter la complexité des équations. Un résultat similaire est obtenu lorsque l'observateur est capable de suivre le point caractéristique tel qu'il apparaîsse immobile centré dans l'image (point de fixation).

Rappelons tout de même que, pour l'ensemble de ces travaux, les résultats obtenus sur les composantes du mouvement de translation le sont à un facteur d'échelle près.

Enfin, n'oublions pas de remarquer la similitude de notre problématique avec celle de la trajectographie bidimensionnelle. Plus particulièrement, on peut citer la thèse d'O. Trémois [Tremois95] qui montre que la localisation d'un mobile nécessite que l'observateur exécute au minimum deux mouvements distincts. Plus précise-

ment, cette thèse s'intéresse au problème de localisation d'un objet sous-marin par une plateforme d'observation utilisant un sonar passif. Tous deux sont supposés se déplacer en mouvement rectiligne uniforme par morceaux.

4.2 Le torseur d'interaction, application à l'estimation du mouvement 3D

Nous avons montré dans le chapitre 2 que les variations d'un motif dans l'image s'écrivait comme la somme des variations dûes à la vitesse de la caméra T_c et de celles dûes au mouvement propre de l'objet :

$$\underline{\dot{s}} = L_{\underline{s}}^T T_c + \frac{\partial \underline{s}}{\partial t}$$

où l'on rappelle que $L_{\underline{s}}^T$ est le torseur d'interaction associé à \underline{s} . De même, la variation de la fonction de tâche s'écrit :

$$\underline{\dot{e}} = \frac{\partial \underline{e}}{\partial r} T_c + \frac{\partial \underline{e}}{\partial t}$$

On peut rappeler que cette dernière relation a été utilisée dans le chapitre précédent pour estimer $\frac{\partial \underline{e}}{\partial t}$, terme introduit après filtrage dans la commande pour compenser les erreurs de traînage induits.

Par ailleurs, en notant la vitesse propre de l'objet T_0 , nous pouvons écrire :

$$\underline{\dot{s}} = L_{\underline{s}}^T T_c - L_{\underline{s}}^T T_0$$

et:

$$\underline{\dot{e}} = \frac{\partial \underline{e}}{\partial r} T_c - \frac{\partial \underline{e}}{\partial r} T_0$$

Pour estimer T_0 , on en déduit :

$$T_0 = -\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right)^+ (\underline{\dot{e}} - \frac{\partial \underline{e}}{\partial \underline{r}} T_c)$$

soit plus exactement (puisque l'on effectue des approximations aussi bien de $\underline{\dot{e}}$ que de $\frac{\partial \underline{e}}{\partial r}$) :

$$T_0 = -\left(\frac{\widehat{\partial e}}{\partial \underline{r}}\right)^+ (\underline{\hat{e}} - \frac{\widehat{\partial e}}{\partial \underline{r}} T_c)$$

soit encore:

$$T_0 = -\left(\frac{\widehat{\partial e}}{\partial \underline{r}}\right)^+ \frac{\widehat{\partial e}}{\partial t}$$

On peut donc relier la vitesse 3D de l'objet à l'estimation de $\frac{\partial \underline{e}}{\partial t}$ effectuée dans le chapitre précédent. Nous verrons cependant que l'ensemble des composantes de T_0 n'est généralement pas observable.

De la même manière, il est possible de relier T_0 à la vitesse propre de l'objet dans l'image. On obtient :

$$T_0 = -L_{\underline{s}}^{T+} \left(\underline{\dot{s}} - L_{\underline{s}}^{T} T_c \right)$$

soit:

$$T_0 = -\widehat{L}_{\underline{s}}^{T^+} \widehat{\frac{\partial \underline{s}}{\partial t}}$$

Ainsi, à partir de l'estimation de $\frac{\partial \underline{s}}{\partial t}$ (ou $\frac{\partial \underline{e}}{\partial t}$), il est possible d'estimer directement les composantes de T_0 n'appartenant pas au noyau de $L^T_{\underline{s}}$ (on néglige ici en outre les perturbations introduites par le modèle choisi pour $L^T_{\underline{s}}$). Par contre, pour les mouvements appartenant au noyau de $L^T_{\underline{s}}$, une ambiguïté totale existe. En effet, ces mouvements, notés T_0^* , sont tels que : $\frac{\partial \underline{s}}{\partial t} = L^T_{\underline{s}} T_0^* = 0$, ce qui rend leur estimation impossible.

Dans le cas des 2 tâches de poursuite que nous avons considérées dans cette thèse (suivi d'un carré et suivi d'un point), on obtient les résultats suivants :

• dans le cas du carré, on rappelle que $L^T_{\underline{s}}$ est de dimension 8×6 et de rang plein 6 ($\frac{\widehat{\partial e}}{\partial \underline{r}} = \mathbb{I}_6$). On peut donc estimer l'ensemble des 6 composantes du mouvement de l'objet. Ce résultat était prévisible, car dans le cas du carré, on établit une liaison rigide entre la caméra et l'objet poursuivi, le mouvement propre de l'objet est donc exactement le même que celui de la caméra.

• dans le cas du point, la matrice d'interaction associée est de dimension 2×6 et de rang plein 2. Le noyau Ker $L_{\underline{s}}^T$ est donc de dimension 4 dans le cas le plus général. Dans le cas où l'on se restreint aux seuls mouvements de translation (puisque des mouvements de rotation d'un point autour de lui-même sont par nature inobservables), on a alors :

$$\operatorname{Ker} L_{\underline{s}}^T = \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

Ce résultat bien connu signifie qu'un mouvement du point dans la direction de sa ligne de vue ne modifie pas sa position dans l'image. Il est donc impossible de remonter à cette composante éventuelle du mouvement de l'objet à partir de la seule observation de $\frac{\partial \underline{s}}{\partial t}$ ou $\frac{\partial \underline{e}}{\partial t}$. De plus, il est également impossible d'obtenir la profondeur du point considéré.

Le cas où l'ensemble du mouvement de l'objet est observable ne présentant pas de difficultés particulières, nous nous focalisons dans la suite de ce chapitre à l'estimation du mouvement 3D d'un point composé d'une vitesse de translation de direction et d'amplitude inconnues.

Le principe général de l'algorithme que nous présentons dans la section suivante est le suivant : nous utilisons deux points de vue distincts de la caméra puisque un seul est insuffisant afin de remonter à l'ensemble du mouvement de l'objet.

On peut finalement noter l'importance de la matrice d'interaction qui permet de relier le mouvement 3D de l'objet (inconnu) au mouvement mesuré dans l'image.

4.3 Estimation du mouvement 3D d'un point

Nous rappelons que la matrice d'interaction calculée dans le cas du point lie le mouvement de la caméra (T_c) , le mouvement de l'objet (T_0) et le mouvement du point dans l'image suivant la relation suivante (voir paragraphe 2.1) :

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \begin{bmatrix} \begin{pmatrix} v_{xcan} \\ V_{ycan} \\ V_{zcan} \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} - \begin{pmatrix} v_{xobj} \\ V_{yobj} \\ V_{zobj} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$
 (4.1)

Les composantes de rotation du mouvement de l'objet sont nulles, puisque l'objet considéré est un point. On remarque donc que les inconnues à estimer sont la profondeur z du point et les composantes de son vecteur vitesse de translation.

Les équations (4.1) peuvent se réécrire sous la forme suivante :

$$\begin{cases}
z\dot{X} = -(V_{xcam} - V_{xobj}) + X(V_{zcam} - V_{zobj}) + (XY\Omega_x - (1 + X^2)\Omega_y + Y\Omega_z)z \\
z\dot{Y} = -(V_{ycam} - V_{yobj}) + Y(V_{zcam} - V_{zobj}) + ((1 + Y^2)\Omega_x - XY\Omega_y - X\Omega_z)z
\end{cases}$$
(4.2)

Rappelons que ces relations sont exprimées dans le repère lié à la caméra. La connaissance de la position et de l'orientation de la caméra à tout instant permet de réécrire ces égalités dans un repère fixe quelconque.

En utilisant deux positions distinctes de la caméra, on obtient alors un système linéaire à 4 équations (2 équations données par (4.2) au changement de repère près pour chaque position de la caméra) et 5 inconnues (3 pour les composantes de la vitesse de l'objet, supposées constantes dans un repère fixe, et 2 pour exprimer la profondeur du point à chaque position considérée). Pour lever l'indeterminée restante et obtenir un système linéaire surdimensionné, nous écrivons la relation de Chasles entre ces deux situations. En effet (voir Figure 4.1), nous avons une configuration géométrique où quatre positions particulières dans l'espace (deux pour la caméra et deux pour le point) peuvent être considérées. Une équation vectorielle est alors simple à trouver.

On a donc:

$$\vec{O_1 M_1} + \vec{M_1 M_2} + \vec{M_2 O_2} = \vec{O_1 O_2} \tag{4.3}$$

En détaillant les termes de cette relation, d'après l'hypothèse de la projection perspective sur le modèle de la caméra, on a :

$$\vec{O_1 M_1} = X_1 z_1 \vec{x_1} + Y_1 z_1 \vec{y_1} + z_1 \vec{z_1} \tag{4.4}$$

$$\vec{O_2 M_2} = X_2 z_2 \vec{x_2} + Y_2 z_2 \vec{y_2} + z_2 \vec{z_2} \tag{4.5}$$

 $\overrightarrow{O_1O_2}$ est connu et est donné par la mesure du déplacement de la caméra. Enfin, $\overrightarrow{M_1M_2}$ est en fait ce qu'on recherche puisque ce vecteur peut s'écrire :

$$\vec{M_1 M_2} = \vec{V_{obj}} \cdot \Delta T \tag{4.6}$$

où ΔT est l'intervalle de temps séparant les deux positions considérées de la caméra. La relation vectorielle fournit un système de trois équations scalaires. On obtient donc au total un système de sept équations à cinq inconnues qu'on résoud par la

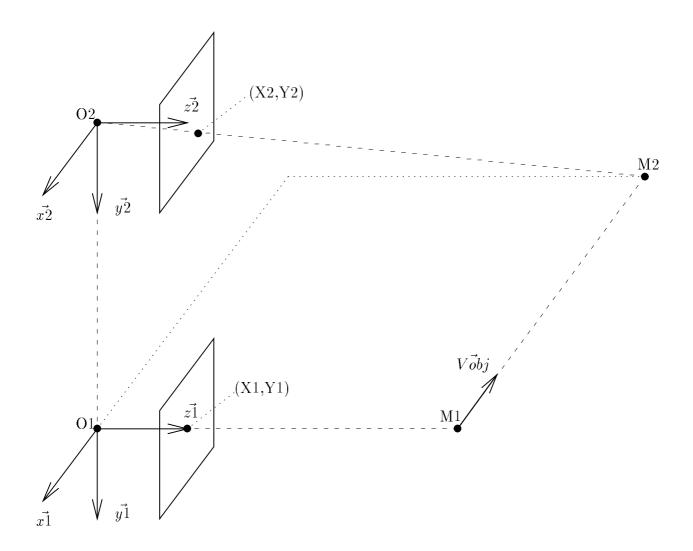


Figure 4.1 : Relation de Chasles

technique des moindres carrés, ce qui nous permet d'estimer la norme de la vitesse 3D du point de même que sa profondeur.

Comme nous le verrons dans les résultats expérimentaux, cette méthode comme toutes les méthodes continues (voir [Boukir93] par exemple) est sujette aux erreurs de discrétisation. En effet, dans les équations (4.2) sont utilisées les mesures de \dot{X} et \dot{Y} . Or cette mesure est en général impossible à réaliser puisque les mesures effectuées dans l'image permettent seulement d'obtenir $\frac{\Delta X}{\Delta t}$ et $\frac{\Delta Y}{\Delta t}$. L'utilisation de ces valeurs à la place de \dot{X} et \dot{Y} entraîne donc des erreurs de discrétisation que l'on peut supprimer dans notre cas en gardant la profondeur du point z constante et en assurant aussi \dot{X} et \dot{Y} constants (on a alors $\frac{\Delta X}{\Delta t} = \dot{X}$ et $\frac{\Delta Y}{\Delta t} = \dot{Y}$). On retrouve là-encore l'intérêt de l'approche vision active qui permet de déterminer les mouvements adéquats de la caméra. La génération automatique de ces mouvements (par asservissement visuel, notamment en utilisant les techniques de suivi développées dans le chapitre précédent) ne sera malheureusement pas abordée dans cette thèse, mais en fait une des plus prometteuses perspectives.

4.3.1 Une mauvaise piste!

Il parait séduisant d'éliminer la variable de profondeur z entre les deux équations du système (4.1). On obtient alors une équation linéaire ne faisant intervenir que la vitesse de l'objet :

$$[XY\Omega_{x} - (1+X^{2})\Omega_{y} + Y\Omega_{z} - \dot{X}][(V_{ycam} - V_{yobj}) - Y(V_{zcam} - V_{zobj})]$$

$$= [(1+Y^{2})\Omega_{x} - XY\Omega_{y} - X\Omega_{z} - \dot{Y}][(V_{xcam} - V_{xobj}) - X(V_{zcam} - V_{zobj})]$$
(4.7)

En écrivant cette relation pour les deux points de vue de la caméra, et en rajoutant la relation de Chasles entre elles, on aboutit alors à un système de cinq équations à cinq inconnues.

Cependant, ce système peut être non résolvable pour certains cas où le système entier de sept équations nous fournit bien la solution. La raison en est qu'en simplifiant les deux équations du système (4.1), nous perdons de l'information, en particulier si l'un des termes en facteur dans l'équation (4.7) est nul.

4.3.2 Quelques cas particuliers

4.3.2.1 Cas d'un objet immobile

Notre schéma d'estimation est robuste au cas particulier d'un objet immobile, ce qui n'est pas le cas si l'on se base sur l'équation (4.7). Dans ce cas, comme nous le verrons dans la section suivante, on remonte à la position 3D du point et sa vitesse nulle est correctement estimée.

4.3.2.2 Cas où les mouvements de la caméra et de l'objet sont coplanaires

L'estimation du mouvement 3D de l'objet est correctement réalisée dans le cas particulier où le ou les mouvements de la caméra et de l'objet sont coplanaires. Il en est de même dans le cas où on utilise deux positions fixes de la caméra formant un même plan avec la trajectoire de l'objet. Ceci montre bien que notre schéma d'estimation est plus puissant qu'une simple intersection de plans d'observation.

4.3.2.3 De l'impossibilité à remonter au mouvement 3D de l'objet dans le cas d'un changement d'orientation seulement

Si l'on fait varier uniquement l'orientation de la caméra alors l'estimation de la vitesse 3D de l'objet ne peut se faire. En effet, le système d'équations (4.2) se réécrit :

$$\begin{cases} z\dot{X} = V_{xobj} - XV_{zobj} + (XY\Omega_x - (1+X^2)\Omega_y + Y\Omega_z)z \\ \\ z\dot{Y} = V_{yobj} - YV_{zobj} + ((1+Y^2)\Omega_x - XY\Omega_y - X\Omega_z)z \end{cases}$$

et la relation de Chasles devient avec $\vec{O_1O_2} = \vec{0}$:

$$(X_1\vec{x_1} + Y_1\vec{y_1} + \vec{z_1}) z_1 + \vec{V_{obj}} \cdot \Delta T - (X_2\vec{x_2} + Y_2\vec{y_2} + \vec{z_2}) z_2 = \vec{0}$$

On vérifie facilement que si on multiplie la profondeur du point et sa vitesse par un même facteur alors on aboutit au même système d'équations, ce qui entraîne une infinité de solutions possibles.

4.3.2.4 De l'impossibilité à remonter au mouvement 3D de l'objet dans le cas d'une caméra animée d'un seul mouvement de translation uniforme

Considérons le repère lié à la caméra (son orientation ne varie pas au cours du temps puisque la caméra n'est animée que d'un mouvement de translation pur) et

un objet situé initialement aux coordonnées $(x_0, y_0, z_0)^T$ et animé d'une vitesse $\vec{V_{obj}}$. La caméra est elle animée d'une vitesse $\vec{V_{cam}}$. Les coordonnées de l'objet exprimées dans le repère caméra seront donc à un instant t1 égales à :

$$(x_1, y_1, z_1)^T = (x_0, y_0, z_0)^T + (\vec{V_{obj}} - \vec{V_{cam}})t_1.$$

Les coordonnées de son image sont $(X_1, Y_1)^T = (x_1/z_1, y_1/z_1)^T$.

Si on considère maintenant un point initialement situé en $(2x_0, 2y_0, 2z_0)^T$ et animé d'une vitesse $\vec{V_{obj2}}$ telle que $\vec{V_{obj2}} - \vec{V_{cam}} = 2(\vec{V_{obj}} - \vec{V_{cam}})$, c'est-à-dire un point deux fois plus profond et avec une vitesse relative par rapport à la caméra double, alors ce point sera à l'instant t_1 situé en :

$$(x_2, y_2, z_2)^T = (2x_0, 2y_0, 2z_0)^T + (\vec{V_{obj2}} - \vec{V_{cam}})t_1 = (2x_1, 2y_1, 2z_1)^T.$$

Les coordonnées de l'image de ce point seront donc les mêmes que celles de l'objet précédent et ceci à tout instant. Il est évident que ceci reste vrai en considérant un autre facteur de proportionnalité que 2.

Ces différentes solutions vérifient bien les équations de notre système. En effet, cela est évident pour les quatres premières équations qui dérivent de la relation du torseur d'interaction. En ce qui concerne, la relation de Chasles, nous pouvons écrire pour le premier objet :

$$\vec{O_1 M_1} + \vec{M_1 M_2} = \vec{O_2 M_2} + \vec{O_1 O_2}$$

Cette relation peut se réécrire après développement :

$$X_1 z_1 \vec{x_1} + Y_1 z_1 \vec{y_1} + z_1 \vec{z_1} + \vec{V_{obj}} \cdot \Delta T = X_2 z_2 \vec{x_2} + Y_2 z_2 \vec{y_2} + z_2 \vec{z_2} + \vec{V_{cam}} \cdot \Delta T$$

La relation de Chasles pour le second objet s'écrit :

$$2X_1z_1\vec{x_1} + 2Y_1z_1\vec{y_1} + 2z_1\vec{z_1} + \vec{V_{obj2}}.\Delta T = 2X_2z_2\vec{x_2} + 2Y_2z_2\vec{y_2} + 2z_2\vec{z_2} + \vec{V_{cam}}.\Delta T$$

En écrivant : $\vec{V_{obj2}} = 2\vec{V_{obj}} - \vec{V_{cam}}$, on remarque que cette dernière égalité est bien équivalente à l'égalité correspondante pour le premier objet.

Ainsi, dans le cas d'un seul mouvement de translation, il existe une infinité de solutions. Pour estimer la vitesse de l'objet, il convient donc soit de faire deux mouvements de translation distincts, soit d'effectuer un mouvement de la caméra qui ne soit pas une simple translation (consignes non nulles pour les rotations).

4.4 Résultats de simulation

Pendant une phase d'initialisation, nous recueillons une série de données qui seront directement exploitées à la fin de cette phase avec les nouvelles données acquises. L'estimation du mouvement 3D de l'objet est donc réalisée en considérant a chaque fois les données recueillies à des instants séparés par un intervalle de temps fixé à l'avance (Figure 4.2).

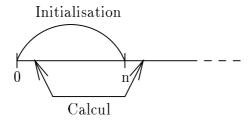


Figure 4.2 : Schéma d'estimation

Dans toute la suite, et pour tous les résultats (de simulation et d'expérimentations réelles) les vitesses seront exprimées en cm/s et les profondeurs en cm. Les composantes du vecteur vitesse de l'objet sont donnés dans un repère fixe (et doivent donc être constantes). Les profondeurs sont, elles, exprimées dans le repère de la caméra.

De plus, la vitesse mesurée dans l'image est mesurée (et approximée) de la manière suivante :

$$\dot{X}_k = \frac{X_k - X_{k-m}}{m \ \Delta t} \ , \ \dot{Y}_k = \frac{Y_k - Y_{k-m}}{m \ \Delta t}$$
 (4.8)

où Δt représente l'intervalle de temps entre 2 acquisitions d'images successives et où m représente l'écart choisi. Sauf cas particulier, nous avons choisi pour l'ensemble des expérimentations une valeur de m égale à 10.

4.4.1 Estimation à partir de deux positions fixes de la caméra

La première expérimentation que nous avons simulée considère deux positions fixes de la caméra. Un parallèle peut alors être effectué par rapport à un système stéréoscopique. Cependant, dans notre cas, les deux points de vue sont réalisés avec la même caméra à des instants différents, ce qui implique que le point a évidemment bougé entre les deux positions utilisées.

L'expérimentation est réalisée comme suit : nous recueillons une série de valeurs durant la première position qui correspond en totalité à une phase d'initialisation, puis après le changement de position de la caméra, nous exploitons les nouvelles données conjointement avec celles de la première position.

4.4.1.1 Cas général, sans bruit

Les erreurs de discrétisation dans le calcul de la vitesse de l'objet dans l'image, évoquées dans le paragraphe précédent, entraînent des erreurs au niveau de l'estimation des composantes du vecteur vitesse de l'objet ainsi que dans l'estimation de la profondeur. Ces erreurs sont très faibles pour la vitesse de l'objet, mais relativement importantes pour la profondeur (de l'ordre du cm).

Les résultats de simulation obtenus en utilisant des données parfaites (c'est-à-dire sans bruit) sont présentés sur les Figures 4.3 et 4.4. Pour cette simulation, la vitesse de l'objet spécifiée est de (5,-3,2), le point étant à environ 100 cm des positions de la caméra. Nous avons également spécifié un déplacement de (10,100,15) entre les deux positions de la caméra.

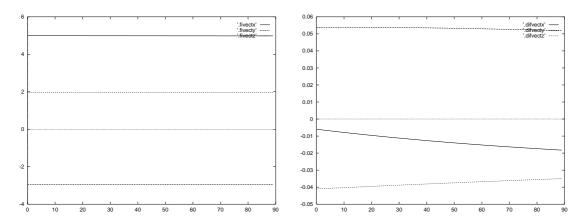


Figure 4.3 : Cas général (avec 2 positions fixes) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

4.4.1.2 Profondeur constante, sans bruit

Si la vitesse de l'objet dans la direction de l'axe optique de la caméra est nulle (ce qui peut être obtenu grâce à une orientation adéquate de la caméra), sa profondeur ne varie pas et les erreurs de discrétisation disparaissent. Nous verrons dans les paragraphes 4.4.3.2 et 4.4.3.3 qu'il est également possible par une autre méthode de réduire ces erreurs (en choisissant une valeur de m plus faible). Malheureusement,

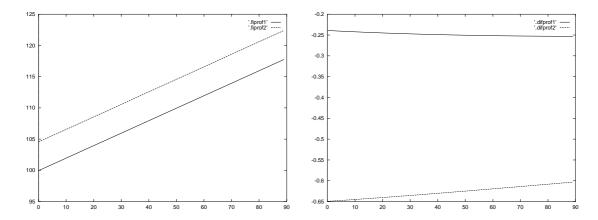


Figure 4.4 : Cas général (avec 2 positions fixes) : Valeurs des profondeurs, erreurs d'estimation

la stabilité et la robustesse des résultats obtenus ne sont alors plus assurées. Dans le cas présent, le fait de conserver une profondeur constante entraîne une estimation exacte des composantes de la vitesse ainsi que de la profondeur, comme on peut le voir sur les Figures 4.5 et 4.6 (Vitesse réelle = (5,-3,0), même déplacement de la caméra que précédemment).

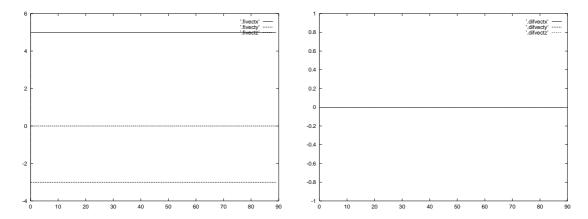


Figure 4.5 : Profondeur constante (avec 2 positions fixes) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

4.4.1.3 Cas général, avec bruit

Nous présentons à présent les résultats obtenus avec un bruit simulé. Précisons que ce bruit porte sur les coordonnées du point dans l'image (bruit uniforme d'amplitude

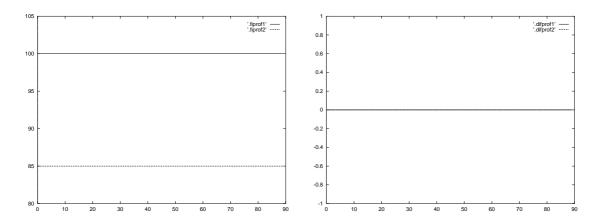


Figure 4.6: Profondeur constante (avec 2 positions fixes): Valeurs des profondeurs, erreurs d'estimation

maximale 1/10 pixel qui correspond à la précision de notre traitement d'image) ainsi que sur la position de la caméra (bruit uniforme d'ampiltude maximale 0.1 cm sur chacune des 3 translations et 0.1 degre sur chacun des 3 angles de rotation qui correspond à la précision de notre système robotique). Nous remarquons que les résultats obtenus sont tout à fait exploitables. Notons qu'il s'agit de résultats bruts, n'ayant subi aucun filtrage ou moyennage. En particulier, et en faisant abstraction des erreurs de discrétisation déjà mises en évidence précédemment, nous remarquons que l'estimation de la vitesse du point est très satisfaisante et que l'erreur sur la profondeur du point ne dépasse pas 1.5 cm. Voir Figures 4.7 et 4.8 (Vitesse réelle = (5,-3,2)).

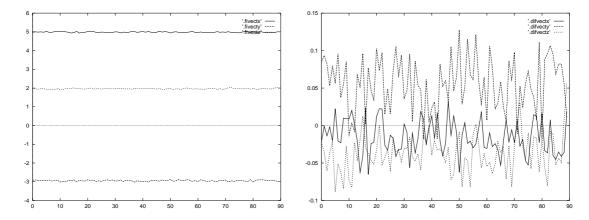


Figure 4.7 : Cas général, avec bruit (avec 2 positions fixes) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

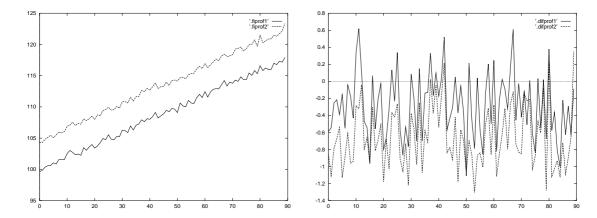


Figure 4.8 : Cas général, avec bruit (avec 2 positions fixes) : Valeurs des profondeurs, erreurs d'estimation

4.4.1.4 Profondeur constante, avec bruit

Le fait de conserver la profondeur du point à la caméra constante fait disparaître les erreurs de discrétisation. Ceci est visible au niveau des erreurs d'estimation qui se trouvent, contrairement au cas précédent, centrées autour de zéro. Voir Figures 4.9 et 4.10 (Vitesse réelle = (5,-3,0)).

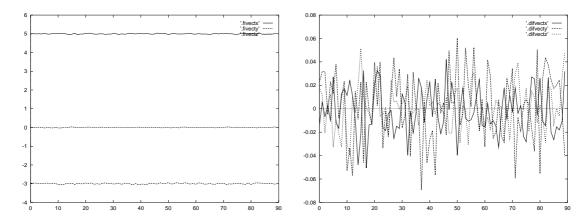


Figure 4.9 : Profondeur constante, avec bruit (avec 2 positions fixes) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

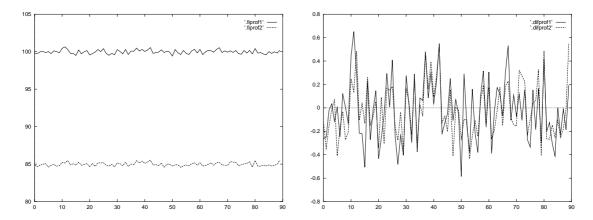


Figure 4.10 : Profondeur constante, avec bruit (avec 2 positions fixes) : Valeurs des profondeurs, erreurs d'estimation

4.4.1.5 Faible déplacement entre les deux positions de la caméra, avec bruit

Pour cette simulation, la distance entre les deux positions de la caméra a été fixée à une valeur plus faible que dans les cas précédents (en l'occurence (5,5,5) à la place de (10,100,15)). Il s'en suit comme prévu une estimation beaucoup plus sensible au bruit. L'erreur sur la vitesse est en effet amplifiée d'un facteur 5 environ (voir Figure 4.11) et l'erreur sur la profondeur d'un facteur supérieur à 10. Nous voyons donc, comme pour toutes les méthodes de reconstruction 3D, l'intérêt d'effectuer un déplacement de la caméra le plus grand possible.

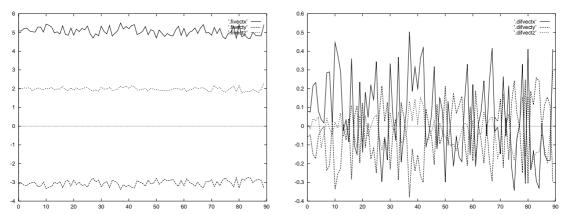


Figure 4.11 : Faible déplacement entre les 2 positions de la caméra, avec bruit (avec 2 positions fixes) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

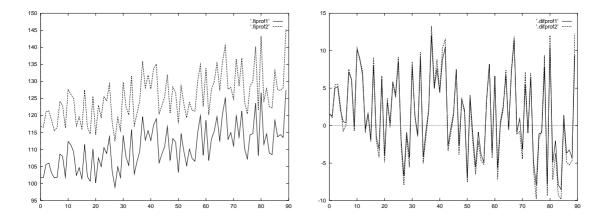


Figure 4.12 : Faible déplacement entre les 2 positions de la caméra, avec bruit (avec 2 positions fixes) : Valeurs des profondeurs, erreurs d'estimation

4.4.1.6 L'objet est à l'arrêt, sans bruit

Comme indiqué précédemment, notre estimation est robuste au cas où l'objet est immobile. Elle estime correctement la valeur nulle du vecteur vitesse, et fournit la profondeur du point. Voir Figures 4.13 et 4.14 (Vitesse réelle = (0,0,0)).

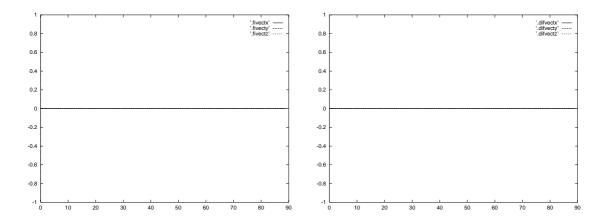


Figure 4.13 : L'objet est à l'arrêt (avec 2 positions fixes) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

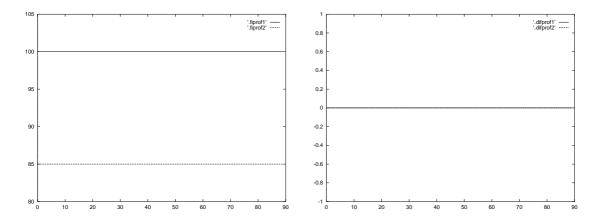


Figure 4.14 : L'objet est à l'arrêt (avec 2 positions fixes) : Valeurs des profondeurs, erreurs d'estimation

4.4.1.7 L'objet est à l'arrêt, avec bruit

Nous présentons sur les Figures 4.15 et 4.16 les résultats obtenus pour la même expérimentation que précédemment, mais cette fois-ci avec du bruit sur les mesures dans l'image ainsi que sur la position de la caméra. Les résultats sont ici encore tout à fait satisfaisants.

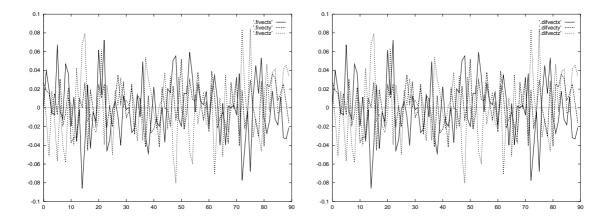


Figure 4.15 : L'objet est à l'arrêt, avec bruit (avec 2 positions fixes) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

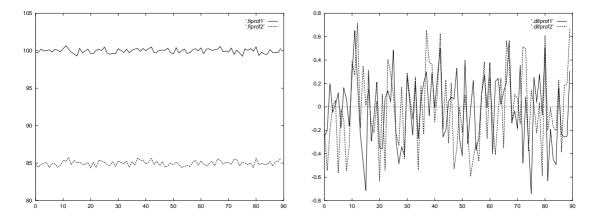


Figure 4.16 : L'objet est à l'arrêt, avec bruit (avec 2 positions fixes) : Valeurs des profondeurs, erreurs d'estimation

4.4.2 Estimation à partir de deux mouvements différents de la caméra

Nous nous intéressons dans ce paragraphe à l'estimation du mouvement 3D d'un objet dans le contexte plus général d'une caméra en mouvement. Dans un premier temps, nous considérons seulement deux mouvements différents de pure translation (rappelons que l'utilisation d'un seul mouvement de translation rend impossible la détermination sans ambiguïté du mouvement et de la profondeur du point). Le cas plus général d'un seul mouvement combinant translation et rotation sera décrit dans le paragraphe 4.2.3 suivant.

Nous initialisons l'algorithme d'estimation pendant le premier mouvement, et ce n'est qu'à partir du moment où la caméra débute son second mouvement que l'estimation du mouvement 3D de l'objet commence. Cette méthode est plus intéressante et plus réaliste que celle présentée précédemment car elle permettrait de contrôler le mouvement de la caméra en fonction du mouvement du point et en particulier de permettre de mieux le suivre et/ou de mieux estimer son mouvement.

4.4.2.1 Cas général, sans bruit

Pour les résultats présentés sur les Figures 4.17 et 4.18 (Vitesse réelle = (5,-3,2)), la caméra exécute deux mouvements de translation successifs, de vitesse respectivement (0,5,0) et (0,-5,0). De nouveau, on remarque l'effet des erreurs de discrétisation sur l'estimation du vecteur vitesse et la profondeur. On peut noter que les erreurs de discrétisation sont beaucoup plus importantes que dans les cas précédents où la

caméra était fixe. Ce résultat est tout à fait normal puisque le mouvement de la caméra introduit un mouvement plus important dans l'image, ce qui entraîne une erreur plus importante entre la valeur de $(\frac{\Delta X}{\Delta t}, \frac{\Delta Y}{\Delta t})$ et la valeur réelle de \dot{X} et \dot{Y} .

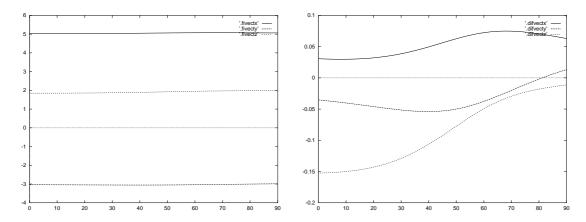


Figure 4.17 : Cas général (avec 2 mouvements) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

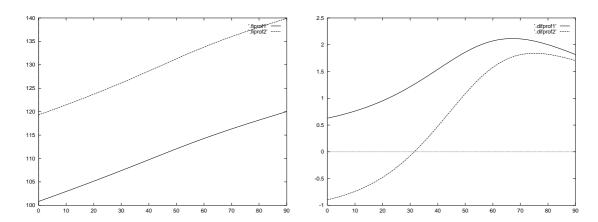


Figure 4.18 : Cas général (avec 2 mouvements) : Valeurs des profondeurs, erreurs d'estimation

4.4.2.2 Profondeur constante, sans bruit

Sur les résultats donnés Figures 4.19 et 4.20 (Vitesse réelle = (5,-3,2)), la caméra exécute deux mouvements de translation successifs, de vitesse respectivement (0,5,2)

et (0,-5,2). La profondeur ne variant pas (vitesse de la caméra dans la direction de l'axe optique égale à celle de l'objet suivant le même axe), on observe une estimation exacte grâce à la disparition des erreurs de discrétisation.

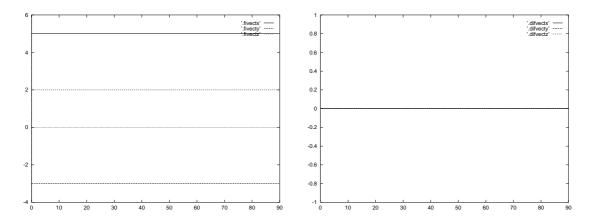


Figure 4.19 : Profondeur constante (avec 2 mouvements) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

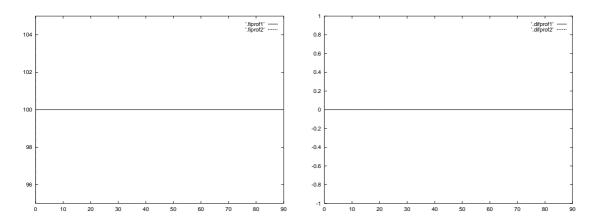


Figure 4.20 : Profondeur constante (avec 2 mouvements) : Valeurs des profondeurs, erreurs d'estimation

4.4.2.3 Cas général, avec bruit

Nous voyons bien sur les Figures 4.21 et 4.22 (Vitesse réelle = (5,-3,2)) qu'avec un taux de bruit correspondant au cas des expérimentations réelles (et identique aux simulations bruitées précédentes), nous obtenons des résultats tout à fait acceptables.

En effet, l'erreur sur la vitesse est de l'ordre du mm/s et sur la profondeur de l'ordre du cm. Rappelons qu'il s'agit de résultats bruts n'ayant subi aucun filtrage.

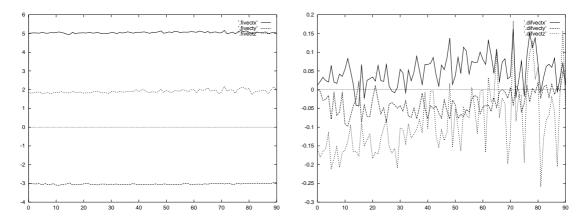


Figure 4.21 : Cas général, avec bruit (avec 2 mouvements) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

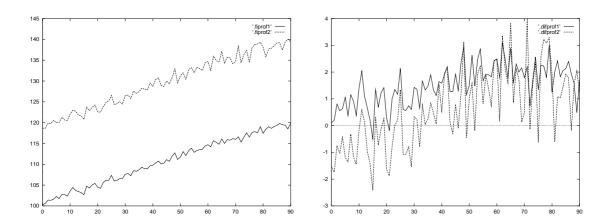


Figure 4.22 : Cas général, avec bruit (avec 2 mouvements) : Valeurs des profondeurs, erreurs d'estimation

4.4.2.4 Profondeur constante, avec bruit

Dans le cas où la profondeur du point reste constante (Voir Figures 4.23 et 4.24 (Vitesse réelle = (5,-3,2))), nous observons de nouveau la suppression des erreurs de discrétisation, puisque les estimations bruitées obtenues sont centrées sur les valeurs exactes.

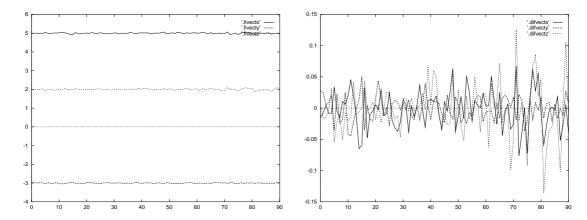


Figure 4.23 : Profondeur constante, avec bruit (avec 2 mouvements) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

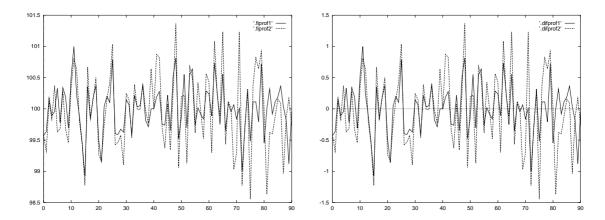


Figure 4.24: Profondeur constante, avec bruit (avec 2 mouvements): Valeurs des profondeurs, erreurs d'estimation

4.4.2.5 L'objet est à l'arrêt, avec bruit

En utilisant deux mouvements de translation, il est également possible d'estimer la profondeur et une vitesse éventuelle nulle de l'objet, comme on peut le voir sur les résultats de simulation bruitées des Figures 4.25 et 4.26.

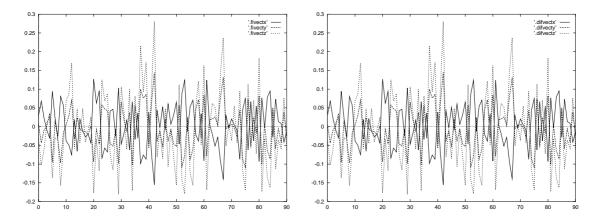


Figure 4.25 : L'objet est à l'arrêt, avec bruit (avec 2 mouvements) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

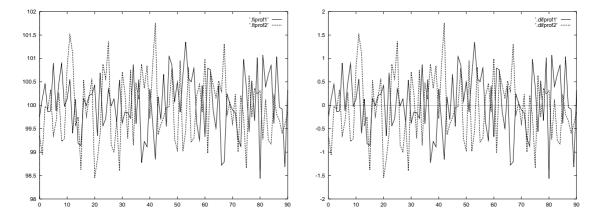


Figure 4.26 : L'objet est à l'arrêt, avec bruit (avec 2 mouvements) : Valeurs des profondeurs, erreurs d'estimation

4.4.3 Estimation à partir d'un seul mouvement de la caméra

Nous étudions ici le cas où la caméra n'exécute qu'un seul mouvement constant en vue de l'estimation du mouvement 3D de l'objet. Comme nous l'avons déjà dit, ce mouvement ne peut se résumer à un mouvement de translation pure mais doit comporter des variations d'orientation. Enfin, comme pour l'ensemble des expérimentations précédentes, l'estimation du mouvement tridimensionnel de l'objet se fait en considérant les données de deux instants séparés par un intervalle de temps choisi à l'avance.

4.4.3.1 Cas général, sans bruit

Nous remarquons que cette méthode, bien que plus intéressante que la précédente dans le sens où on ne fait qu'un seul mouvement, est beaucoup plus sensible aux erreurs de discrétisation. Nous observons sur les Figures 4.27 et 4.28 (Vitesse réelle = (5,-3,2)) que l'estimation du vecteur vitesse et de la profondeur s'écarte assez nettement de sa valeur réelle. Pour cette simulation, la vitesse de la caméra était de (0,5,0) et (0.05,0,0) rd/s. Là encore, ce résultat était (malheureusement) attendu, puisque l'introduction de composantes de mouvement de rotation augmente les erreurs entre $(\frac{\Delta X}{\Delta t}, \frac{\Delta Y}{\Delta t})$ et (\dot{X}, \dot{Y}) .

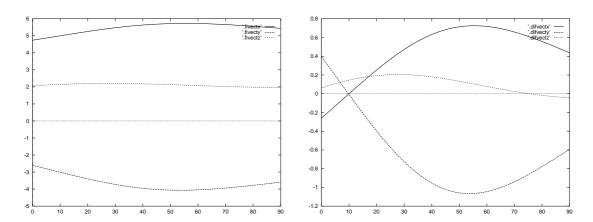


Figure 4.27 : Cas général (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

4.4.3.2 Cas général, vitesse dans l'image calculée sur 1 itération, sans bruit

En absence de bruit, nous démontrons ici que les écarts observés dans l'expérimentation précédente sont bien dûs aux erreurs de discrétisation. Pour cela, nous avons refait l'expérimentation précédente en considérant une seule itération pour le calcul

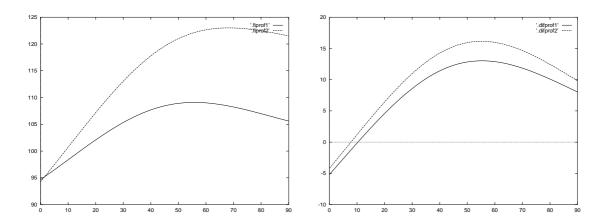


Figure 4.28 : Cas général (avec un seul mouvement) : Valeurs des profondeurs, erreurs d'estimation

de la vitesse dans l'image. En d'autres termes, pour cette simulation, \dot{X} et \dot{Y} sont mesurés de la manière suivante : $\dot{X}_k = \frac{X_k - X_{k-1}}{\Delta t}$, $\dot{Y}_k = \frac{Y_k - Y_{k-1}}{\Delta t}$ Rappelons qu'une valeur de dix itérations a été considérée dans toutes les expé-

Rappelons qu'une valeur de dix itérations a été considérée dans toutes les expérimentations précédentes (voir équation (4.8)). Dans le cas présent, la valeur utilisée pour la vitesse du point dans l'image est nettement plus proche de sa valeur réelle. Du coup, nous observons sur les Figures 4.29 et 4.30 que les résultats sont bien meilleurs que précédemment, les erreurs sur la vitesse et la profondeur étant réduites d'un facteur 10 environ.

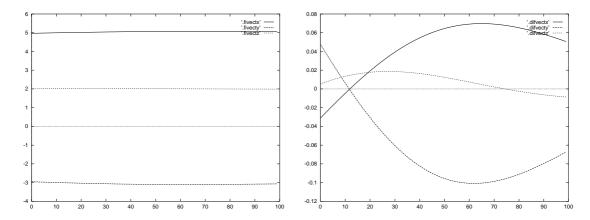


Figure 4.29 : Cas général, vitesse dans l'image calculée sur 1 itération (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

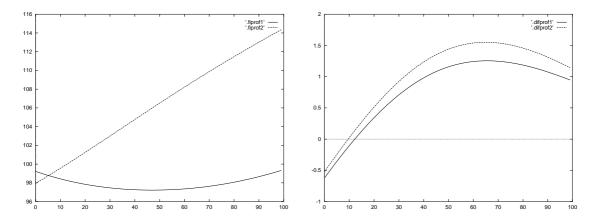


Figure 4.30 : Cas général, vitesse dans l'image calculée sur 1 itération (avec un seul mouvement) : Valeurs des profondeurs, erreurs d'estimation

4.4.3.3 Cas général, vitesse dans l'image calculée sur 1 itération, avec bruit

Malheureusement, la méthode de réduction de l'intervalle de temps pour le calcul de la vitesse du point dans l'image n'est pas réellement applicable. En effet, on s'aperçoit sur les Figures 4.31 et 4.32 qu'elle fournit en milieu bruité des estimations totalement instables, non robustes et parfois aberrantes, allant même jusqu'à donner des profondeurs négatives!

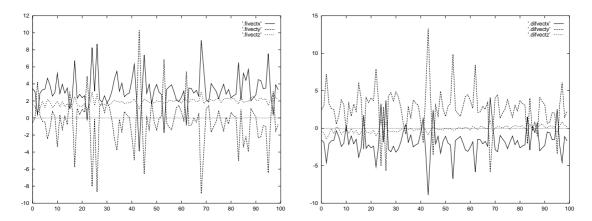


Figure 4.31 : Cas général, vitesse dans l'image calculée sur 1 itération, avec bruit (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

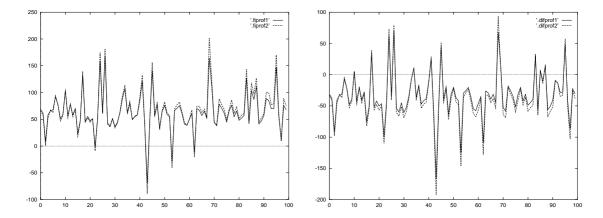


Figure 4.32 : Cas général, vitesse dans l'image calculée sur 1 itération, avec bruit (avec un seul mouvement) : Valeurs des profondeurs, erreurs d'estimation

4.4.3.4 Cas général, avec bruit

Cette expérimentation a été réalisée en présence de bruit, et en considérant un intervalle de 10 itérations pour le calcul de la vitesse du point dans l'image. On observe alors (Voir Figures 4.33 et 4.34) que les résultats sont nettement plus robustes que dans la simulation précédente (gain d'un facteur 10 environ et disparition des estimations aberrantes). Nous remarquons aussi que les résultats sont les mêmes que ceux du cas non bruité (première expérimentation décrite sur les Figures 4.27 et 4.28) sur lesquels vient se superposer du bruit.

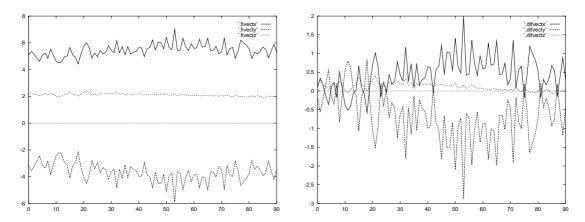


Figure 4.33 : Cas général, avec bruit (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

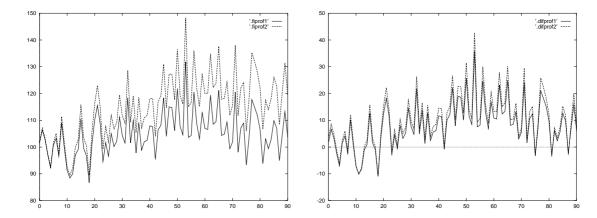


Figure 4.34 : Cas général, avec bruit (avec un seul mouvement) : Valeurs des profondeurs, erreurs d'estimation

4.4.3.5 Cas général, faible déplacement entre O_1 et O_2 , avec bruit

Nous avons repris la même expérimentation que précédemment, sauf qu'on considère ici un intervalle plus court entre les 2 points de vue considérés de la caméra (en l'occurence, un intervalle de 50 itérations au lieu de 100 a été fixé). Il s'en suit de façon logique une détérioration des différentes estimations (Voir Figures 4.35 et 4.36). Nous en déduisons de nouveau l'intérêt d'opérer un déplacement le plus grand possible. Ce résultat est similaire à celui concernant la reconstruction 3D par triangulation et qui stipule de prendre une base de triangulation la plus grande possible pour contrer l'effet des bruits sur les mesures.

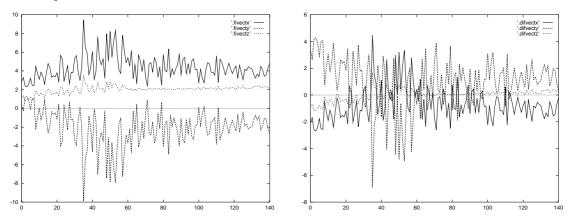


Figure 4.35 : Cas général, faible déplacement entre O_1 et O_2 , avec bruit (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

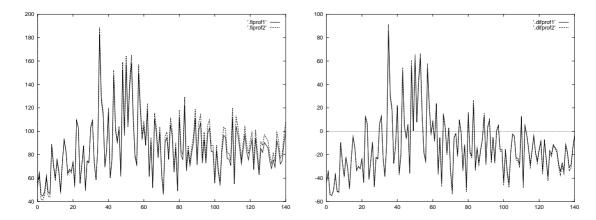


Figure 4.36 : Cas général, faible déplacement entre O_1 et O_2 , avec bruit (avec un seul mouvement) : Valeurs des profondeurs, erreurs d'estimation

4.4.3.6 Profondeur constante, sans bruit

Comme indiqué précédemment, la méthode pour éliminer les conséquences facheuses des erreurs de dicrétisation consiste à garder la profondeur de la caméra à l'objet constante. Ceci est réalisé si la caméra effectue un mouvement d'hélice d'axe de révolution l'axe de mouvement de l'objet (voir Figure 4.38). Les résultats donnés sur les Figures 4.37 et 4.39 (Vitesse réelle = (5,0,0)) montrent le bien-fondé d'une telle démarche (l'intervalle pour le calcul de la vitesse dans l'image étant égal à 10 itérations).

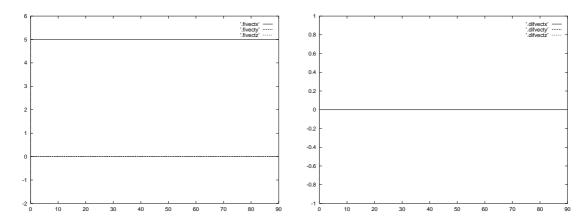


Figure 4.37 : Profondeur constante (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

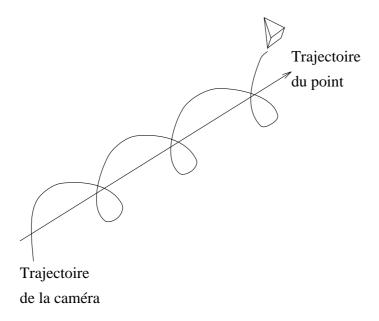


Figure 4.38 : Trajectoire de la caméra

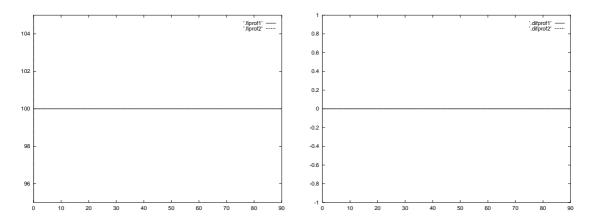


Figure 4.39: Profondeur constante (avec un seul mouvement): Valeurs des profondeurs, erreurs d'estimation

4.4.3.7 Profondeur constante, point immobile au centre de l'image, avec bruit

Contrairement à la solution de réduction de l'intervalle de calcul des vitesses dans l'image, la solution développée précédemment s'applique bien au cas bruité. On peut en effet observer sur les Figures 4.40 et 4.41 (Vitesse réelle = (5,0,0)) que les résultats (bruités bien sûr) restent ici centrés sur les valeurs réelles. Ils donneraient donc après filtrage de bonnes estimées contrairement au cas général.

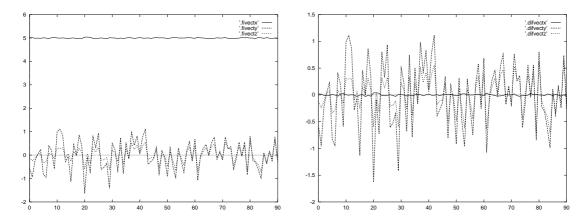


Figure 4.40 : Profondeur constante, point immobile au centre de l'image, avec bruit (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

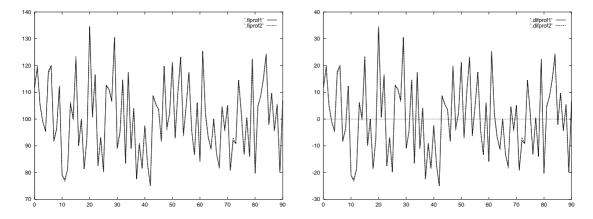


Figure 4.41 : Profondeur constante, point immobile au centre de l'image, avec bruit (avec un seul mouvement) : Valeurs des profondeurs, erreurs d'estimation

4.4.3.8 Quelle position du point dans l'image pour une bonne estimation de son mouvement ?

En gardant la profondeur constante comme dans le cas précédent, nous avons refait l'expérimentation précédente mais en donnant une vitesse à la caméra de telle sorte que le point soit mobile dans l'image (Voir Figures 4.42 et 4.43), puis avec le point immobile dans l'image mais non au centre de celle-ci (Voir Figures 4.44 et 4.45).

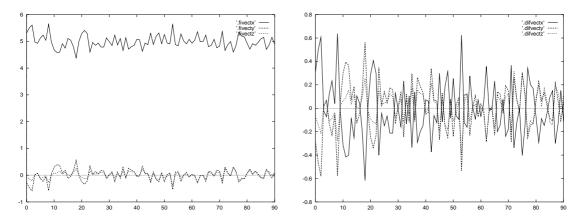


Figure 4.42 : Profondeur constante, point mobile dans l'image, avec bruit (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

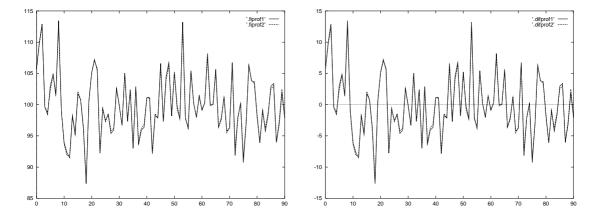


Figure 4.43: Profondeur constante, point mobile dans l'image, avec bruit (avec un seul mouvement): Valeurs des profondeurs, erreurs d'estimation

Nous observons que la meilleure configuration pour obtenir une bonne estimation du mouvement du point est de le garder immobile loin du centre de l'image. Ce

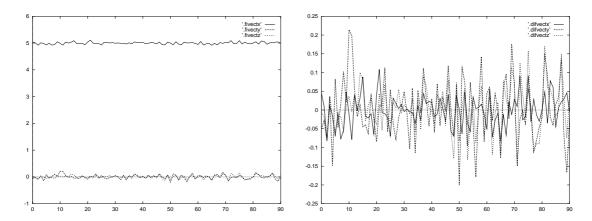


Figure 4.44 : Profondeur constante, point immobile (pas au centre de l'image), avec bruit (avec un seul mouvement) : Valeurs des composantes du vecteur vitesse de l'objet, erreurs d'estimation

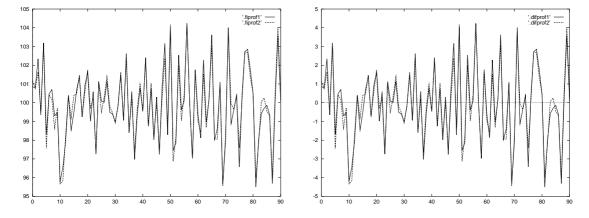


Figure 4.45 : Profondeur constante, point immobile (pas au centre de l'image), avec bruit (avec un seul mouvement) : Valeurs des profondeurs, erreurs d'estimation

résultat est contradictoire avec ceux de [Boukir93] et [Viéville et al.96b] qui ont démontré qu'on obtenait une reconstruction 3D plus robuste si un point était fixé au centre de l'image sur la zone fovéale. Cependant, dans ces travaux, l'application était évidemment différente puisque les scènes considérées sont fixes, alors que dans notre cas, on souhaite remonter au mouvement 3D d'un objet mobile. Par contre, le fait de garder le point immobile dans l'image (et avec une profondeur constante) est tout à fait similaire aux résultats de S. Boukir pour supprimer les erreurs de discrétisation.

On peut finalement signaler que ce type de mouvement permet également d'estimer une éventuelle vitesse nulle de l'objet. Les résultats sont similaires à ceux obtenus en utilisant deux positions fixes ou deux mouvements de translation.

4.5 Résultats expérimentaux

Nous avons réalisé des expérimentations réelles sur notre cellule expérimentale précédemment décrite. Les résultats obtenus se sont avérés très satisfaisants et ont validé notre approche pour l'estimation du mouvement 3D de l'objet. Toutes les expérimentations ont été réalisées avec un point animé d'un mouvement rectiligne uniforme se déplaçant à une vitesse d'amplitude 3 cm/s.

4.5.1 Estimation à partir de deux positions fixes de la caméra

Avec deux positions fixes de la caméra séparées d'une distance d'approximativement 60 cm, nous obtenons les résultats de la Figure 4.46. Nous pouvons vérifier que la norme du vecteur vitesse estimé est bien égale à 3 cm/s. On peut également remarquer que les résultats sont extrêmement stables et robustes.

Nous avons par la suite modifié la trajectoire du point de sorte qu'on s'approche le plus possible de la configuration "profondeur constante". Les résultats correspondants sont montrés sur la Figure 4.47.

4.5.2 Estimation à partir de deux mouvements différents de la caméra

Nous avons gardé la même direction initiale de mouvement pour le point. La caméra exécute ici un double mouvement de translation assez lent (+1.5 cm/s pendant 7 secondes suivi d'un mouvement de même amplitude mais de sens opposé pendant 7 autres secondes). Les résultats obtenus sont corrects (Voir Figure 4.48). Précisons

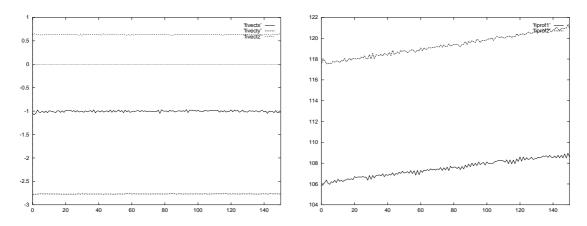


Figure 4.46 : Résultats expérimentaux (avec 2 positions fixes) : Composantes du vecteur vitesse de l'objet, Profondeurs

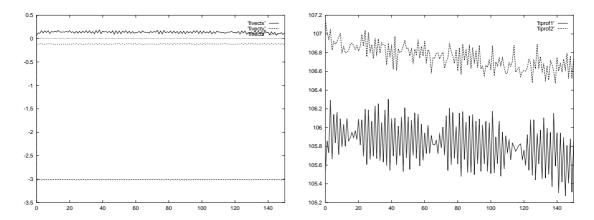


Figure 4.47 : Résultats expérimentaux (avec 2 positions fixes) : Composantes du vecteur vitesse de l'objet, Profondeurs

qu'il est tout à fait normal que les profondeurs calculées ici soient différentes de celles obtenues avec 2 positions fixes puisque exprimées dans le repère caméra. Les résultats sont légèrement plus bruités que dans le cas précédent (ce qui avait été mis en évidence dans les résultats de simulation), mais restent tout à fait satisfaisants : erreur de l'ordre du mm/s sur la vitesse et inférieure au cm pour la profondeur.

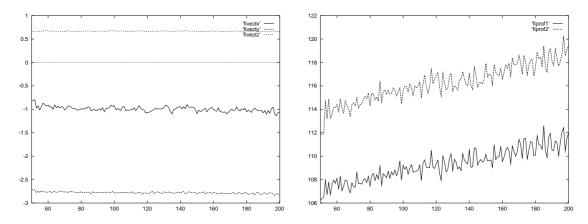


Figure 4.48 : Résultats expérimentaux (avec 2 mouvements) : Composantes du vecteur vitesse de l'objet, Profondeurs

4.5.3 Estimation à partir d'un seul mouvement de la caméra

Toujours avec la même trajectoire spatiale du point que précédemment, nous estimons sa vitesse en exécutant un mouvement de grande amplitude de type hélice. Plus précisément, la vitesse de la caméra à été fixée à (3,-5,0) cm/s et (-2.7,0,0) dg/s. Les résultats obtenus sont plus bruités que pour les deux autres méthodes (voir Figure 4.49). Ceci est logique en raison des erreurs de discrétisation plus importantes (la profondeur du point ne restant pas constante au cours de l'expérimentation).

En considérant la configuration "profondeur constante". Les résultats obtenus montrent une amélioration très nette de l'estimation (voir Figure 4.50). L'erreur sur le mouvement étant de l'ordre du mm/s. L'erreur sur la profondeur est de l'ordre de quelques cm, mais rappelons qu'un simple moyennage (par exemple sur 3 itérations successives) permettrait d'améliorer encore très nettement ces résultats.

Pour des expérimentations plus réalistes, il faudrait générer la commande par asservissement visuel en vue de respecter les critères d'optimalité et de robustesse au bruit de l'estimation du mouvement.

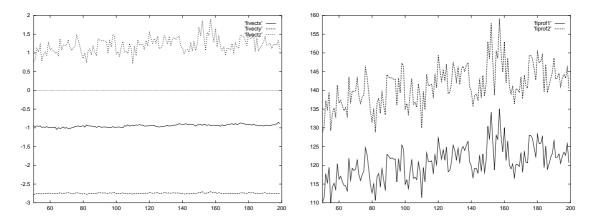


Figure 4.49 : Résultats expérimentaux (avec un seul mouvement) : Composantes du vecteur vitesse de l'objet, Profondeurs

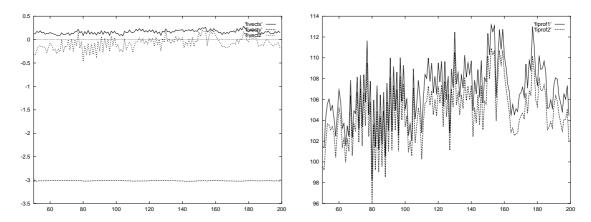


Figure 4.50 : Résultats expérimentaux (avec un seul mouvement) : Composantes du vecteur vitesse de l'objet, Profondeurs

Conclusion 129

4.6 Conclusion

Nous avons proposé un schéma d'estimation du mouvement tridimensionnel d'un point. Les méthodes développées ont été validées en simulation et par des expérimentations réelles. Les résultats obtenus sont très satisfaisants sachant qu'aucun filtrage des résultats n'a été effectué.

Signalons que contrairement aux travaux déjà réalisés dans le domaine, la connaissance du mouvement de la caméra permet de lever la classique ambiguïté entre l'amplitude de la vitesse et la profondeur.

Nous avons trouvé quelques résultats pour une bonne estimation, à savoir : considérer des points de vue les plus distants possible, et garder une profondeur du point constante par rapport à la caméra.

Nous avons observé dans le cas d'un seul mouvement de la caméra, que le fait de garder l'objet immobile loin du centre de l'image donne l'estimation la moins sensible au bruit. Il reste cependant à le démontrer théoriquement. Cependant, ce résultat ne doit pas nous faire oublier que l'image de l'objet ne doit pas être non plus "trop" loin du centre de l'image, faute de quoi, le risque de le perdre est plus grand.

Conclusion 131

Conclusion

Nous nous sommes intéressés dans cette thèse aux aspects de suivi et d'estimation du mouvement d'un objet. Ces aspects sont étroitement liés, le bon déroulement de l'un implique la bonne réalisation de l'autre.

Nous sommes partis d'un savoir-faire, développé à l'IRISA, et concernant la génération de la commande à appliquer à un robot muni d'une caméra pour effectuer des tâches de positionnement.

Dans un premier stade, nous avons introduit dans la commande cinématique un gain adaptatif au lieu d'un gain constant. Ceci a pour effet immédiat d'obtenir un gain important au niveau du temps de convergence pour des tâches de positionnement. Nos travaux ont permis aussi un calcul automatique des paramètres de ce gain adaptatif à partir de la connaissance des gains maximaux admissibles.

Nous nous sommes intéressés par la suite aux tâches de suivi. Nous avons développé un schéma pour l'estimation du mouvement 2D. Cette estimation est nécessaire si l'on veut éviter le problème des erreurs de traînage. Un travail important a concerné le filtrage de cette estimation, celle-ci étant bruitée, et ceci en respectant plusieurs hypothèses de mouvement, y compris des variations brusques de la vitesse ou de l'accélération de l'objet.

Une suite intéressante à ce travail serait l'organisation d'une banque de filtres [Viéville et al.95]. On sélectionnerait le filtre adéquat au type de mouvement de l'objet à partir des mesures dans l'image et des estimées obtenues.

Dans une seconde étape, nous nous sommes focalisés sur l'estimation du mouvement 3D. Un schéma d'estimation a été développé puis validé par simulation et expérimentations réelles. Des résultats ont été démontrés concernant le mouvement

132 Conclusion

optimal à faire effectuer à la caméra. D'autres ont été seulement observés. Leur véritable adéquation à une estimation stable et robuste mériterait d'être démontrée. Une autre perspective serait de respecter les conditions trouvées en générant la commande correspondante par asservissement visuel. En effet, une perspective prometteuse serait de combiner le suivi et l'estimation du mouvement 3D. Il est tentant de trouver les moyens de faire profiter les tâches de suivi de la connaissance du mouvement 3D de l'objet. On peut aisément imaginer, que dans le sens inverse, le bon suivi profite à l'estimation du mouvement 3D en gardant par exemple l'objet immobile dans l'image.

Annexe A

Initialisation des filtres de Kalman

Nous décrivons dans cette annexe l'initialisation des différents paramètres du filtre de Kalman, notamment la variance de l'erreur d'estimation initiale. Nous avons adopté la même méthode que celle suivie dans [Hunt et al.82]. Elle consiste, après avoir fixé les variances des bruits d'état et de mesure, à calculer les erreurs initiales d'estimation du vecteur d'état et à en déduire la valeur de la matrice de covariance associée.

A.1 Vitesse constante - bruit blanc

Les équations du filtre de Kalman sont données dans ce cas par :

$$\begin{cases} x_{k+1} = x_k + v_k \\ z_k = x_k + w_k \end{cases}$$

La variance du bruit d'état est notée σ_Q^2 et la variance du bruit de mesure σ_R^2 . On se donne pour estimée initiale de la vitesse la première mesure effectuée, soit :

$$\hat{x}_1 = z_1$$

et on en déduit l'erreur sur l'estimée de la vitesse :

$$\tilde{x}_1 = \hat{x}_1 - x_1 = z_1 - x_1 = w_1$$

La variance de l'erreur d'estimation est donc :

$$P_{(0|0)} = \sigma_R^2$$

A.2 Accélération constante - bruit blanc

Dans ce cas, les équations du filtre de Kalman sont les suivantes :

$$\begin{cases} x_{k+1} = x_k + \Delta \dot{x}_k + v_{1k} \\ \dot{x}_{k+1} = \dot{x}_k + v_{2k} \\ z_k = x_k + w_k \end{cases}$$

La matrice de covariance du bruit d'état est donnée par :

$$Q = \begin{pmatrix} \sigma_{Q1}^2 & 0 \\ 0 & \sigma_{Q2}^2 \end{pmatrix}$$

et la variance du bruit de mesure par σ_R^2 . On choisit pour estimée initiale de la vitesse et de l'accélération :

$$\hat{x}_2 = z_2$$
 , $\hat{x}_2 = \frac{z_2 - z_1}{\Lambda}$

On en déduit l'erreur sur l'estimée de la vitesse et de l'accélération :

$$\tilde{x}_2 = \hat{x}_2 - x_2 = z_2 - x_2 = w_2$$
, $\tilde{x}_2 = \hat{x}_2 - \dot{x}_2 = \frac{z_2 - z_1}{\Delta} - \dot{x}_2$

On peut exprimer \dot{x}_2 à partir du modèle d'état. En effet, on a :

$$\begin{cases} x_2 = x_1 + \Delta \dot{x}_1 + v_{11} \\ \dot{x}_2 = \dot{x}_1 + v_{21} \end{cases}$$

ce qui donne :

$$\dot{x}_2 = \frac{(x_2 - x_1) + (\Delta v_{21} - v_{11})}{\Delta}$$

D'autre part, à partir de l'équation d'observation, on a :

$$z_2 - z_1 = (x_2 - x_1) + (w_2 - w_1)$$

L'erreur sur l'estimée de l'accélération s'écrit donc :

$$\tilde{\dot{x}}_2 = \frac{(x_2 - x_1) + (w_2 - w_1)}{\Lambda} - \frac{(x_2 - x_1) + (\Delta v_{21} - v_{11})}{\Lambda}$$

ou encore:

$$\dot{\tilde{x}}_2 = \frac{w_2 - w_1 + v_{11} - \Delta v_{21}}{\Delta}$$

A partir des expressions de \tilde{x}_2 et $\tilde{\dot{x}}_2$, on en déduit la matrice de covariance initiale :

$$P_{(0|0)} = \begin{pmatrix} \sigma_R^2 & \frac{\sigma_R^2}{\Delta} \\ \\ \frac{\sigma_R^2}{\Delta} & \frac{2\sigma_R^2 + \sigma_{Q1}^2}{\Delta^2} + \sigma_{Q2}^2 \end{pmatrix}$$

A.3 Vitesse constante - bruit corrélé

Dans ce cas, les équations du filtre de Kalman sont données par :

$$\begin{cases} x_{k+1} = x_k + \nu_k \\ \nu_{k+1} = \rho \nu_k + \nu_k \\ z_k = x_k + w_k \end{cases}$$

La matrice de covariance du bruit d'état a maintenant la forme suivante :

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & \sigma_O^2 \end{pmatrix}$$

et la variance du bruit de mesure est encore notée σ_R^2 .

Comme dans le cas précédent du filtre à vitesse constante, l'estimée initiale de la vitesse est naturellement choisie égale à la valeur de la première mesure effectuée :

$$\hat{x}_1 = z_1$$

et l'estimée du bruit corrélé est choisie nulle :

$$\hat{\nu}_1 = 0$$

L'erreur sur l'estimée de la vitesse et du bruit corrélé est donc :

$$\tilde{x}_1 = \hat{x}_1 - x_1 = z_1 - x_1 = w_1$$
, $\tilde{\nu}_1 = \hat{\nu}_1 - \nu_1 = -\nu_1$

On en déduit :

$$P_{(0|0)} = \begin{pmatrix} \sigma_R^2 & 0 \\ & & \\ 0 & \sigma_1^2 \end{pmatrix}$$

où σ_1^2 est la variance du bruit corrélé, soit $\sigma_1^2 = \frac{\sigma_Q^2}{1 - \rho^2}$.

A.4 Accélération constante - bruit corrélé

Pour ce dernier cas, les équations du filtre de Kalman sont données par :

$$\begin{cases} x_{k+1} = x_k + \nu_k + \Delta \dot{x}_k \\ \nu_{k+1} = \rho \nu_k + \nu_{1k} \\ \dot{x}_{k+1} = \dot{x}_k + \nu_{2k} \\ z_k = x_k + w_k \end{cases}$$

avec pour matrice de covariance du bruit d'état :

$$Q = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sigma_{Q1}^2 & 0 \\ 0 & 0 & \sigma_{Q2}^2 \end{pmatrix}$$

et pour variance du bruit de mesure σ_R^2 .

L'estimée initiale de la vitesse, du bruit corrélé et de l'accélération sont fixées de la manière suivante :

$$\hat{x}_2 = z_2$$
 , $\hat{\nu}_2 = 0$, $\hat{x}_2 = \frac{z_2 - z_1}{\Delta}$

ce qui donne :

$$\tilde{x}_2 = z_2 - x_2 = w_2$$
 , $\tilde{\dot{x}}_2 = \frac{(w_2 - w_1) + \nu_1}{\Delta} - v_{21}$, $\tilde{\nu}_2 = -\nu_2$

car $\tilde{\dot{x}}_2$ s'écrit :

$$\tilde{\dot{x}}_2 = \frac{z_2 - z_1}{\Delta} - (\dot{x}_1 + v_{21}) = \frac{(x_2 - x_1) + (w_2 - w_1)}{\Delta} - \frac{(x_2 - x_1) - \nu_1}{\Delta} - v_{21}$$

La matrice de covariance initiale est donc :

$$P_{(0|0)} = \begin{pmatrix} \sigma_R^2 & 0 & \frac{\sigma_R^2}{\Delta} \\ 0 & \sigma_1^2 & -\frac{\rho \sigma_1^2}{\Delta} \\ \\ \frac{\sigma_R^2}{\Delta} & -\frac{\rho \sigma_1^2}{\Delta} & \frac{2\sigma_R^2 + \sigma_1^2}{\Delta^2} + \sigma_{Q2}^2 \end{pmatrix}$$

où σ_1^2 est la variance du bruit corrélé, soit $\sigma_1^2 = \frac{\sigma_{Q1}^2}{1-\rho^2}$

Bibliographie

[Adiv85]	Adiv (G). – Determining three-dimensional motion and structure from optical flow generated by several moving objects. <i>IEEE Trans. on PAMI</i> , vol. 7, n° 4, pp. 384–401, Juillet 1985.
[Aggarwal et al.87]	Aggarwal (J.K.) et Wang (Y.F.). – Analysis of a sequence of images using point and line correspondences. Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1275–1280, Mars 1987.
[Aggarwal et al.88]	Aggarwal (J.K.) et Nandhakumar (N.). – On the computation of motion from sequences of images, a review. Proc. of the IEEE, vol. 76, n° 8, pp. 917–935, Août 1988.
[Allen et al.91]	Allen (P.K.), Yoshimi (B.) et Timcenko (A.). – Realtime visual servoing. <i>Proc. of IEEE Int. Conf. on Robotics and Automation</i> , pp. 851–856, Sacramento, USA, Avril 1991.
[Allen et al.93]	Allen (P.K.), Timcenko (A.), Yoshimi (B.) et Michelman (P.). – Hand-eye coordination for robotic tracking and grasping. <i>Visual Servoing</i> , éd. par Hashimoto (K.), pp. 33–69, World Scientific Series in Robotics and Automated Systems, vol 7, World Scientific Press, Singapore, 1993.
[Aloimonos et al.87]	Aloimonos (J.), Weiss (I.) et Bandopadhay (A.). – Active vision. <i>Proc. of 1st Int. Conf. on Computer Vision</i> , pp. 35–54, Londres, Juin 1987.
[Aloimonos et al.91]	Aloimonos (J.) et Tsakiris (D.). – On the visual mathematics of tracking. <i>Image and Vision Computing</i> ,

138 Bibliographie

vol. 9, n° 4, pp. 235–251, Août 1991.

[Ballard 91] Ballard (D.H.). – Animate vision. Artificial Intelligence, 48:57–86, 1991.

[Bandopadhay et al.86] Bandopadhay (A.), Chandra (B.) et Ballard (D.). – Egomotion using active vision. *Int. Conf. on CVPR* 86, pp. 498–503, Juin 1986.

[Bar-Shalom et al.93] Bar-Shalom (Y.) et Li. (X.R.). – Estimation and tracking: principles, techniques and software. *Artech House*, Boston, 1993.

Basseville (M.) et Benveniste (A.). – Design and comparative study of some sequential jump detection algorithms for digital signals. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 31, n° 3, pp. 521–535, Juin 1983.

Behringer (R). – Detection of discontinuities of road curvature by glr methods. *Proc. of Int. Symposium on Intelligent Vehicles*, Detroit, USA, Septembre 1995.

Bensalah (F.) et Chaumette (F.). – Commande de l'orientation d'une caméra embarquée sur un robot sousmarin par asservissement visuel. Rapport technique, convention INRIA/IFREMER, Novembre 1993.

Bensalah (F.) et Chaumette (F.). – Détection de rupture de modèles appliquée à l'asservissement visuel. Rapport technique n° 886, IRISA, Décembre 1994.

Bensalah (F.) et Chaumette (F.). – Real time visual tracking using the generalized likelihood ratio test. *Proc. of Int. Conf. on Automation, Robotics and Computer Vision*, pp. 1379–1383, Singapore, Novembre 1994.

Bensalah (F.) et Chaumette (F.). – Compensation of abrupt motion changes in target tracking by visual servoing. *IROS'95*, pp. 181–187, Pittsburgh, USA, Août 1995.

[Basseville et al.83]

[Behringer95]

[Bensalah et al.93]

[Bensalah et al.94a]

[Bensalah et al.94b]

[Bensalah et al.95a]

[Bensalah et al.95b] Bensalah (F.) et Chaumette (F.). – Poursuite de cible par asservissement visuel avec test du rapport de vraisemblance généralisé. ICIA '95, pp. 343-348, Nancy, Juin 1995. [Blackman86] Blackman (S.). - Multiple Target Tracking with Radar Application. Artech House Inc., 1986. [Bobet94] Bobet (P.). – Tête stéréoscopique, réflexes oculaires et vision. Thèse de l'Institut National Polytechnique de Grenoble, Décembre 1994. [Boukarri89] Boukarri (B.). – Reconstruction 3D récursive de scènes structurées au moyen d'une caméra mobile. Application à la robotique. Thèse de l'Université d'Orsay, Octobre 1989. [Boukir93] Boukir (S.). - Reconstruction 3D d'un environnement statique par vision active. Thèse de l'Université de Rennes I, Octobre 1993. Broida (T.) et Chellapa. (R.). – Estimation of object [Broida et al.86] motion parameters from noisy images. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 8, n° 1, pp. 90-99, Janvier 1986. Broida (T.J.) et Chellappa (R.). – Estimating the kine-[Broida et al.91] matics and structure of a rigid object from a sequence of monocular images. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 13, n° 6, pp. 497–513, Juin 1991. [Brown92] Brown (C.). - Issues in selective perception. - IAPR Int. Conf. on Pattern Recognition, vol. 1, pp. 21-30, La Have, Pays Bas, Août 1992. Brunnström (K.), Eklundh (J.O.) et Uhlin (T.). - Ac-[Brunnström et al.96]

[Chaumette et al.91]

tive fixation for scene exploration. International Journal

Chaumette (F.), Rives (P.) et Espiau (B.). – Positioning of a robot with respect to an object, tracking it and

of Computer Vision, 17(2):137-162, Février 1996.

estimating its velocity by visual servoing. *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2248–2253, Sacramento, USA, Avril 1991.

[Chaumette et al.93]

Chaumette (F.) et Santos (A.). – Tracking a moving object by visual servoing. *Proc. of 12th IFAC World Congress*, pp. 409–414, Sydney, Australia, Juillet 1993.

[Chaumette90]

Chaumette (F.). – La relation vision-commande : théorie et application à des tâches robotiques. *Thèse de l'Université de Rennes I*, Juillet 1990.

[Christy et al.96]

Christy (S.) et Horaud (R.). – Euclidean reconstruction : from paraperspective to perspective. Lecture Notes in Computer Science n° 1065, ECCV'96, vol. 2, pp. 129–140, Cambridge, Avril 1996.

[Cipolla et al.92]

Cipolla (R.) et Blake (A.). – Surface orientation and time to contact from image divergence and deformation. *ECCV'92*, pp. 187-202, Gênes, Italie, 1992.

[Corke et al.92]

Corke (P.I.) et Good (M.C.). – Dynamics effects in high performance visual servoing. *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1838–1843, Mai 1992.

[Corke94]

Corke (P.I.). – High-Performance Visual Closed-Loop Robot Control. *Thèse de PhD*, University of Melbourne, Juillet 1994.

[Crowley et al.95]

Crowley (J.L.), Mesrabi (M.) et Chaumette (F.). – Comparison of kinematic and visual servoing for fixation. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 335–341, Pittsburgh, USA, Août 1995.

[Doignon et al.94]

Doignon (C.), Abba (G.) et Ostertag (E.). – Recognition and localization of solid objects by a monocular vision system for robotic tasks. *IROS'94*, Munich, Septembre 1994.

[Doignon et al.95] Doignon (C.), Werghi (N.), Abba (G.) et Ostertag (E.).

Reconnaissance et localisation d'objets en robotique par un système de vision monoculaire. *ICIA '95*, pp. 511–516, Nancy, Juin 1995.

Dornaika (F.). – Contribution à l'intégration vision/robotique : calibration, localisation, asservissement. *Thèse de l'INPG*, Grenoble, Septembre 1995.

Espiau (B.), Chaumette (F.) et Rives (P.). – A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, vol. 8, n° 3, pp. 313–326, Juin 1992.

Espiau (B.). – Effect of camera calibration errors on visual servoing in robotics. *Third Int. Symposium on Experimental Robotics*, Kyoto, Japan, Octobre 1993.

Espiau (B.). – Asservissement Visuel et Commande de la Distance Focale. *Rapport de recherche Inria*, n° 2613, Juillet 1995.

Faugeras (O.D.), Lustman (F.) et Toscani (G). – Motion and structure from motion from point and line matches. *Proc. of 1st Int. Conf. on Computer Vision*, pp. 25–34, Londres, Juin 1987.

Faugeras (O.) et Maybank (S.). – Motion from point matches : multiplicity os solutions. *Rapport de recherche INRIA*, n° 1157, Sophia-Antipolis, Février 1990.

Feddema (J.T.) et Lee (C.S.G.). – Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, n° 5, pp. 1172–1183, Octobre 1990.

Feddema (J.T), Lee (C.S.G.) et Mitchell (O.R.). – Weighted selection of image features for resolved rate visual feedback control. *IEEE Trans. on Robotics and Automation*, vol. 7, n° 1, pp. 31–47, Février 1991.

[Espiau et al.92]

[Dornaika95]

[Espiau93]

[Espiau95]

[Faugeras et al.87]

[Faugeras et al.90]

[Feddema et al.90]

[Feddema et al.91]

[Feddema et al.93]

Feddema (J.T), Lee (C.S.G.) et Mitchell (O.R.). – Feature-Based Visual Servoing of Robotic Systems. *Visual Servoing*, éd. par Hashimoto (K.), pp. 105–138, World Scientific Series in Robotics and Automated Systems, vol 7, World Scientific Press, Singapore, 1993.

[Feddema89]

Feddema (J.T.). – Real-Time Visual Feedback Control for Hand-Eye Coordinated Robotic Systems. *Thèse de PhD, Purdue University*, Août 1989.

[Funda95]

Funda (J.). – Book review: Visual servoing. *IEEE Trans. on Robotics and Automation*, vol. 11, n° 3, pp. 474–475, Juin 1995.

[Germain et al.94]

Germain (F.) et Skordas (T.). – An image motion estimation technique based on a combined statiscal test and spatiotemporal generalised likelihood ratio approach. *Proc. of European Conf. on Computer Vision*, pp. 152–157, Stockholm, Sweden, Mai 1994.

[Hager94]

Hager (G.). – Real-time feature tracking and projective invariance as a basis for hand-eye coordination. *Proc. of CVPR'94*, pp. 533-539, Seattle, USA, Juin 1994.

[Hashimoto et al.93]

Hashimoto (K.) et Kimura (H.). – LQ Optimal and Nonlinear Approaches to Visual Servoing. Visual Servoing, éd. par Hashimoto (K.), pp. 165–198. World Scientific Series in Robotics and Automated Systems, vol 7, World Scientific Press, Singapore, 1993.

[Hashimoto93]

Hashimoto (K. editor). – Visual Servoing: Real Time Control of Robot manipulators based on visual sensory feedback. World Scientific Series in Robotics and Automated Systems, vol 7, World Scientific Press, Singapore, 1993.

[Hinkley71]

Hinkley (D.V.). – Inference about the change-point from cumulative sum tests. *Biometrika*, vol. 58, n° 3, pp. 509–523, Décembre 1971.

[Hunt et al.82] Hunt (A.E.) et Sanderson (A.C.). – Vision-Based Pre-

[Liu et al.86]

dictive Robotic Tracking of a Moving Object. Rapport technique, Carnegie-Mellon University, n° CMU-

RI-TR-82-15, Janvier 1982.

[Khadraoui et al.95] Khadraoui (D.), Motyl (G.), Martinet (P.), Gallice (J.)

et Chaumette (F.). – Visual Servoing in Robotics Scheme using a Camera/Laser-Stripe Sensor. *Rapport*

technique IRISA, n° 898, Janvier 1995.

[Korn et al.82] Korn (J.), Gully (S.W.) et Willsky (A.S.). – Application of the generalized likelihood ratio algorithm to maneu-

ver detection and estimation. Proc. of American Control

Conference, pp. 792–798, Washington DC, USA, 1982.

Liu (Y.C.) et Huang (T.S.). – Estimation of rigid body motion using straight line correspondences: further results. *Proc. of Int. Conf. of Pattern Recognition*,

pp. 306-309, Octobre 1986.

[Mesrabi94] Mesrabi (M.). – Contrôle du Regard pour un Systeme

de Vision Active. Thèse de l'Institut National Polytech-

nique de Grenoble, 1994.

[Meyer 93] Meyer (F.). – Suivi de régions et analyse des trajectoires

dans une séquence d'images. Thèse de l'Université de

Rennes 1, IRISA, Juin 1993.

[Michel et al.93] Michel (H.) et Rives (P.). – Singularities in the Deter-

mination of the Situation of a Robot Effector from the Perspective View of Three Points. Rapport de recherche,

INRIA, Février 1993.

[Milios et al.93] Milios (E.), Jenkin (M.) et Tsosos (J.). – Design and

performance of trish, a binocular robot head with torsional eye movements. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(1):51-68, Fé-

vrier 1993.

[Mitiche et al.89] Mitiche (A.) et Habelrih (G.). – Interpretation of

straight line correspondences using angular relations.

Pattern Recognition, vol. 22, n° 3, pp. 299–308, 1989.

[Motyl92] Motyl (G.). – Couplage d'une caméra et d'un faisceau laser en commande référencée vision. Thèse de l'Université Blaise-Pascal de Clermont-Ferrand, Septembre 1992. [Murray et al.95] Murray (D.), Bradshaw (K.), Mc Lauchlan (P.) et Sharkey (P.). – Driving saccade to pursuit using image motion. International Journal of Computer Vision, 16(3):205-228, Mars 1995. Nelson (B.) et Khosla (P.K.). – Increasing the tracking [Nelson et al.93a] region of an eye-in-hand system by singularity and joint limit avoidance. Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 418-423, Mai 1993. [Nelson et al.93b] Nelson (B.), Papanikolopoulos (N.P.) et Khosla (P.K.). - Visual servoing for robotic assembly. Visual Servoing, éd. par Hashimoto (K.), pp. 139-164. World Scientific Series in Robotics and Automated Systems, vol 7, World Scientific Press, Singapore, 1993. [Papanikolopoulos et al.93] Papanikolopoulos (N.P.), Khosla (P.K.) et Kanade (T.).

- Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. IEEE Trans. on Robotics and Automation, vol. 9, n° 1, pp. 14–35, Février 1993.

[Pissard-Gibollet93] Pissard-Gibollet (R.). - Conception et commande par asservissement visuel d'un robot mobile. Thèse de l'Ecole des Mines de Paris, Sophia-Antipolis, Décembre 1993.

[Rives et al.92] Rives (P.) et Pissard-Gibollet (R.). – Reactive mobile robots based on a visual servoing approach. 13^{rd} Conf. on Artificial Intelligence, Simulation and Planning, Perth, Australia, Juillet 1992.

[Rizzo95]

Rizzo (A.). – Evitement des butées articulaires et des singularités internes en asservissement visuel. Rapport technique, Juin 1995.

[Samson et al.91] Samson (C.), Espiau (B.) et Le Borgne (M.). – Robot Control: the Task Function Approach. Clarendon Press, Oxford, England, 1991. [Santos et al.92] Santos (A.) et Chaumette (F.). – Target Tracking by Visual Servoing. Rapport technique n° 683, IRISA, 1992. [Sharma et al.95] Sharma (R.) et Hutchinson (S.). – Optimizing hand/eye configuration for visual-servo systems. *IEEE Int Conf.* on Robotics and Automation, pp. 172–177, Nagoya, Japon, 1995. [Spindler93] Spindler (F.). – Réalisation de démonstrations en asservissement visuel. Rapport technique, IRISA, Septembre 1993. [Subbarao89] Subbarao (M.). – Interpretation of image flow: a spatiotemporal approach. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 11, n° 3, pp. 266-278, 1989. [Suh93] Suh (I.H.). - Visual servoing of robot manipulators by fuzzy membership function based neural networks. Visual Servoing, éd. par Hashimoto (K.), pp. 285-315. World Scientific Series in Robotics and Automated Systems, vol 7, World Scientific Press, Singapore, 1993. Sundareswaran (V.), Chaumette (F.) et Bouthemy (P.). [Sundareswaran et al.94] Visual servoing using image motion information. Proc. of the IEEE Int. Workshop on Visual Behaviors, pp. 102-106, Seattle, USA, Juin 1994. [Sundareswaran et al.96] Sundareswaran (V.), Bouthemy (P.) et Chaumette (F.). - Exploiting image motion for active vision in a visual servoing framework. Int. Journal of Robotics Research, à paraître vol. 15, n° 6, Décembre 1996. Thompson (B.), Lechleider (P.) et Stuck (E.R.). – De-[Thompson et al.93] tecting moving objects using the rigidity constraint. IEEE Trans. on PAMI, vol. 15, n° 2, pp. 162-166, Fé-

vrier 1993.

[Tomasi et al.92]

Tomasi (C.) et Kanade (T.) – Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, vol. 9, n° 2, pp. 137-154, Novembre 1992.

[Tremois95]

Trémois (O.). – Etude de méthodes de trajectographie pour des sources manœuvrantes. Thèse de l'Université de Rennes I, Juin 1995.

[Tsai et al.81]

Tsai (R.) et Huang (T.). – Estimating three-dimensional motion parameters of a rigid planar patch. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 29, n° 6, pp. 1147-1152, Décembre 1981.

[Tsai et al.84]

Tsai (R.) et Huang (T.). – Uniqueness and estimation of three-dimensional motion parameters of rigid objetcs with curved surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, n° 1, pp. 13-27, Janvier 1984.

[Tsosos92]

Tsosos (J.K.). – On the relative complexity of active v.s. passive visual search. *International Journal of Computer Vision*, 7(2):127–141, 1992.

[Urban et al.94]

Urban (J.-P.), Motyl (G.) et Gallice (J.). – Real-time visual servoing using controlled illumination. *International Journal of Robotics Research*, vol. 13, n° 1, pp. 93–100, Février 1994.

[Urban90]

Urban (J.-P.). – Une approche de type asservissement visuel appliquée à la robotique. *Thèse de l'Université Blaise-Pascal de Clermont-Ferrand*, Juin 1990.

[Viéville et al.95]

Viéville (T.), Clergue (E.), Enciso (R.) et Mathieu (H.).

– Experimenting with 3D vision on a robotic head.

Robotics and Autonomous Systems, vol. 14, pp. 1–27, 1995.

[Viéville et al.96a]

Viéville (T.), Faugeras (O.) et Luong (Q-T.). – Motion of points and lines in the uncalibrated case. *International Journal of Computer Vision*, vol. 17, pp. 7–41, 1996.

[Viéville et al.96b] Viéville (T.) et Lingrand (D.). – Using singular displacements for uncalibrated monocular visual systems.

ECCV'96, vol. 2, pp. 207–216, Cambridge, Avril 1996.

[Weiss et al.87] Weiss (L.E.), Sanderson (A.C.) et Neuman (C.P.). –
Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, vol. 3,

n° 5, pp. 404–417, Octobre 1987.

[Weng et al.92] Weng (J.), Huang (T.S.) et Ahuja (N.). – Motion and structure from line correspondences: closed-form solu-

tion, uniqueness, and optimization. $\it IEEE\ Trans.\ on$

PAMI, vol. 14, n° 3, pp. 318–336, Mars 1992.

[Weng et al.89] Weng (J.), Huang (T.S.) et Ahuja (N.). – Motion and structure from two perspective views: algorithms, error

analysis, and error estimation. *IEEE Trans. on PAMI*,

vol. 11, n° 5, pp. 451–476, Mai 1989.

[Willsky et al.76] Willsky (A.S.) et Jones (H.L.). – A generalized like-

lihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Trans. on Automatic*

Control, vol. 21, n° 1, pp. 108–112, Février 1976.

[Wu et al.88] Wu (J.J.), Rink (R.E.), Caelli (T.M.) et Gourishankar (V.G.). – Recovery of 3-D location and motion of a rigid

object through camera image (an extended Kalman filter approach). Int. Journal of Computer Vision, vol. 3,

pp. 373–394, 1988.

Résumé

Ce travail s'inscrit dans le domaine de l'utilisation de la vision pour la réalisation de tâches robotiques et concerne plus précisément les aspects de poursuite de cible par asservissement visuel et d'estimation du mouvement par vision active.

Nous avons tout d'abord amélioré la réalisation des tâches de positionnement par une détermination pratique des paramètres d'un gain adaptatif. Par la suite, nous nous sommes intéressés au cas d'objets mobiles, et plus particulièrement à l'estimation du mouvement de ces objets, nécessaire pour une exécution satisfaisante des tâches de poursuite. La qualité de poursuite est en effet grandement améliorée grâce à une estimation robuste du mouvement. Celle ci est réalisée en utilisant la technique de détection de rupture de modèle dite du GLR, en parallèle avec un filtre de Kalman.

Nous avons aussi abordé le problème classique en vision par ordinateur concernant l'estimation du mouvement 3D. Nous proposons une approche nouvelle, basée sur l'aspect actif du mouvement de la caméra, permettant de remonter complètement (sans ambiguïté d'échelle) au mouvement tridimensionnel d'un point mobile. Des résultats concernant le mouvement à faire effectuer à la caméra en vue d'une détermination optimale (la moins sensible aux différents bruits) du mouvement spatial du point sont présentés. Plusieurs résultats de simulation viennent illustrer notre méthode.

L'ensemble des méthodes présentées dans cette thèse ont été validées par des expérimentations réalisées sur la cellule de vision robotique de l'Irisa.