

La commande des robots manipulateurs

W. Khalil

Traité IC2, Editions Hermès, 2002

Chapitre 3

Asservissement visuel

Résumé

Ce chapitre est consacré à la présentation des concepts de base des techniques d'asservissement visuel. Il s'agit d'élaborer des lois de commande à partir des informations visuelles fournies par un capteur de vision. La modélisation de ces informations, qui peuvent être de nature extrêmement variée, y tient un rôle particulièrement important, de même que leur sélection et leur intégration au sein de fonctions de tâche à définir correctement en fonction des degrés de liberté à contrôler et des propriétés de stabilité, voire de découplage, souhaitées. L'utilisation d'autres capteurs extéroceptifs est également abordée.

Mots-clés : Asservissement visuel, matrice d'interaction, fonction de tâche

Keywords : Visual servoing, interaction matrix, task function

Table des matières

3	Asservissement visuel	
	<i>F. Chaumette</i>	105
3.1.	Introduction	105
3.2.	Modélisation des informations visuelles	107
3.2.1.	Matrice d'interaction	107
3.2.2.	Cas d'une caméra embarquée	108
3.2.3.	Cas d'une caméra déportée	109
3.2.4.	Matrice d'interaction d'informations 2D	110
3.2.5.	Matrice d'interaction d'informations 3D	118
3.3.	Fonction de tâche et commande	124
3.3.1.	Obtention de la consigne s^*	124
3.3.2.	Régulation de la fonction de tâche	125
3.3.3.	Tâches hybrides	135
3.3.4.	Poursuite de cibles	141
3.4.	Autres capteurs extéroceptifs	142
3.5.	Conclusion	143
3.6.	Remerciements	144
3.7.	Bibliographie	145
	Index	151

Chapitre 3

Asservissement visuel

3.1. Introduction

Les capteurs extéroceptifs dont sont de plus en plus fréquemment dotés les systèmes robotiques fournissent, par définition, des informations sur l'environnement dans lequel ils évoluent. Leur utilisation est bien évidemment indispensable dès qu'il s'agit d'accomplir des tâches dans des environnements non parfaitement rigides ou connus. Ils permettent également de considérer d'éventuelles erreurs ou imprécisions dans l'identification des modèles géométriques (et donc cinématiques) des robots manipulateurs. Outre les capteurs d'effort dont l'intérêt et l'exploitation ont été vus dans le chapitre précédent, on dispose de capteurs fournissant une localisation du système dans son environnement ou une perception plus ou moins locale de celui-ci [RIV 01a, RIV 01b]. A titre d'exemple, les systèmes de marquage au sol, les systèmes basés sur des balises passives ou radio, ainsi que les récents GPS, permettent de localiser un robot mobile en déterminant soit sa position absolue soit son déplacement. Côté perception, les capteurs proximétriques fournissent une information sur la distance aux objets les plus proches. Ils sont donc particulièrement adaptés aux tâches d'évitement d'obstacles. Les systèmes de vision par ordinateur et de télémétrie jouent quant à eux un rôle assez particulier puisqu'ils peuvent être employés aussi bien pour des besoins de localisation, de navigation, et d'exploration.

La reconstruction 3D a longtemps été considérée comme un module préalable, indispensable et indépendant à tout module de génération de mouvement d'un robot dans un environnement imparfaitement connu. En vision par ordinateur, cet état de fait, justifié par les temps de calcul prohibitifs des algorithmes de traitements d'images, a donné lieu à de nombreux et fructueux travaux, dans le domaine de la stéréovision notamment [FAU 93]. Les progrès algorithmiques

et technologiques ont permis depuis une quinzaine d'années de lier plus étroitement les aspects de perception et d'action, en intégrant directement les mesures fournies par un système de vision dans des lois de commande en boucle fermée sur les informations visuelles extraites. Cette approche, dite d'asservissement visuel, rejoint les travaux sur la commande référencée capteurs et fait l'objet de ce chapitre.

Les techniques d'asservissement visuel consistent à utiliser les informations fournies par une ou plusieurs caméras afin de contrôler les mouvements d'un système robotique. On peut ainsi réaliser une grande variété de tâches de positionnement du système par rapport à son environnement, ou de poursuite d'objets mobiles, en contrôlant de un à l'ensemble des n degrés de liberté du système. Quelle que soit la configuration du capteur, pouvant aller d'une caméra embarquée sur l'effecteur du robot à plusieurs caméras déportées, il s'agit de sélectionner au mieux un ensemble de k mesures, permettant de contrôler les m degrés de liberté souhaités, et d'élaborer une loi de commande afin que ces mesures $\mathbf{s}(t)$ atteignent une valeur désirée \mathbf{s}^* , définissant une réalisation correcte de la tâche. On peut également vouloir suivre une trajectoire $\mathbf{s}^*(t)$. Le principe de la commande se ramène donc à réguler (c'est-à-dire amener et maintenir à zéro) le vecteur d'erreur $\mathbf{s}(t) - \mathbf{s}^*(t)$.

Avec un capteur de vision, fournissant à la base des informations 2D, la nature des mesures potentielles est extrêmement riche, puisque l'on peut considérer en asservissement visuel aussi bien des mesures 2D, telles que les coordonnées de points caractéristiques dans l'image par exemple, que des mesures 3D, fournies par un module de localisation exploitant les informations 2D extraites (voir figure 3.1). De cette richesse provient la difficulté majeure de l'asservissement visuel, à savoir, parmi l'ensemble des informations potentielles, comment sélectionner celles qui fourniront un comportement satisfaisant au système. Les qualités souhaitées sont nombreuses : stabilité locale voire globale, robustesse aux erreurs de mesure et aux erreurs de modélisation, absence de singularités et de minima locaux, trajectoires satisfaisantes du robot, mais aussi des informations visuelles dans l'image, et enfin découplage maximal entre les informations visuelles et les degrés de liberté commandés.

Pour étudier le comportement du système obtenu, il est nécessaire de modéliser la relation qui lie la variation des mesures choisies $\mathbf{s}(t)$ aux variables de contrôle. Cette phase primordiale de modélisation est décrite à présent. Par contre, nous n'aborderons pas dans ce chapitre les aspects de traitements d'images, indispensables pour extraire d'une image numérique les informations 2D utiles, et les suivre à chaque itération de la loi de commande. Nous renvoyons le lecteur intéressé à des ouvrages spécialisés dans ce domaine [HOR 95, VIN 00].

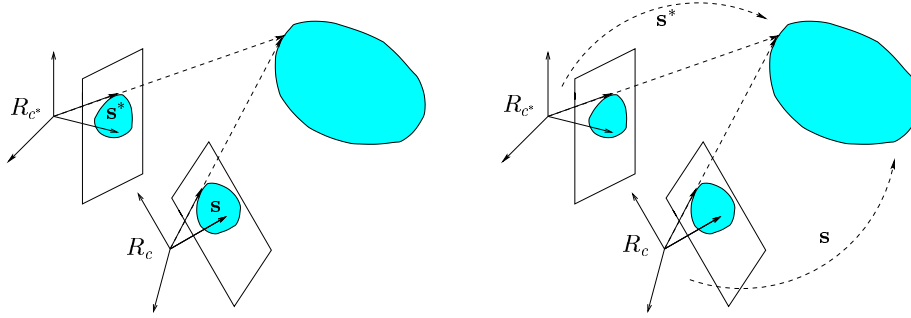


Figure 3.1. Asservissements visuels 2D et 3D : pour amener le repère de la caméra de R_c en R_{c^*} , on utilise en asservissement visuel 2D des mesures \mathbf{s} extraites directement de l'image (à gauche). En asservissement visuel 3D, les mesures \mathbf{s} représentent des informations 3D estimées à l'issue de la phase de traitement d'image (à droite).

3.2. Modélisation des informations visuelles

3.2.1. Matrice d'interaction

Un ensemble \mathbf{s} de k informations visuelles peut être pris en compte dans un schéma d'asservissement visuel à partir du moment où il est défini par la donnée d'une application différentiable de SE_3 dans \mathbb{R}^k :

$$\mathbf{s} = \mathbf{s}(\mathbf{r}(t)) \quad [3.1]$$

où $\mathbf{r}(t)$, élément de l'espace SE_3 de représentation des repères et des corps rigides, décrit la situation à l'instant t entre la caméra et son environnement. On considère donc que seuls des mouvements de la caméra ou des objets qu'elle perçoit sont susceptibles de faire varier la valeur d'une information visuelle.

La différentielle de \mathbf{s} permet de relier les variations des informations visuelles au mouvement relatif entre la caméra et la scène. On obtient en effet en dérivant [3.1] :

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \dot{\mathbf{r}} = \mathbf{L}_s \mathcal{T} \quad [3.2]$$

où :

- \mathbf{L}_s est une matrice jacobienne de dimension $k \times 6$, appelée la **matrice d'interaction** associée à \mathbf{s} .
- \mathcal{T} est le torseur cinématique relatif entre la caméra et la scène, exprimé dans le repère de la caméra R_c en son origine C . Plus précisément, si \mathcal{T}_c et \mathcal{T}_o sont respectivement les torseurs cinématiques de la caméra et de la scène qu'elle perçoit, tous deux exprimés dans R_c et en C , on pose :

$$\mathcal{T} = \mathcal{T}_c - \mathcal{T}_o \quad [3.3]$$

Par la suite, et sauf exception, on écrira par abus de langage qu'un torseur exprimé dans un repère a sa valeur donnée en l'origine de ce repère. On note par ailleurs \mathbf{V} la vitesse de translation à l'origine du repère et ω la vitesse de rotation telles que $\mathcal{T} = (\mathbf{V}^T \omega^T)^T$. Si ${}^o\mathbf{R}_c$ décrit la matrice de rotation entre un repère R_o lié à l'objet et R_c , on a par définition [SAM 91] :

$$\tilde{\omega} = {}^o\dot{\mathbf{R}}_c {}^o\mathbf{R}_c^T = -{}^c\dot{\mathbf{R}}_o {}^c\mathbf{R}_o^T = {}^c\mathbf{R}_o {}^o\dot{\mathbf{R}}_c \quad [3.4]$$

où $\tilde{\omega}$ est la matrice antisymétrique du préproduit vectoriel associé à ω .

REMARQUE.— De manière plus formelle [SAM 91], la transposée de la matrice d'interaction peut être définie comme la représentation matricielle du sous-espace engendré par une famille de k torseurs exprimés dans R_c . En effet, la différentielle de chaque composante de \mathbf{s} se décompose sous la forme du produit de deux torseurs, l'un dit d'interaction, l'autre étant bien entendu le torseur cinématique. Nous verrons l'intérêt pratique de cette définition au paragraphe 3.3.3.1.

3.2.2. Cas d'une caméra embarquée

Si l'on considère une caméra embarquée sur l'effecteur d'un robot manipulateur observant un objet immobile, le lien entre $\dot{\mathbf{s}}$ et la vitesse des variables articulaires $\dot{\mathbf{q}}$ du robot s'obtient aisément :

$$\dot{\mathbf{s}} = \mathbf{L}_s {}^c\mathbb{T}_n {}^n\mathbf{J}_n(\mathbf{q}) \dot{\mathbf{q}} \quad [3.5]$$

où :

- ${}^n\mathbf{J}_n(\mathbf{q})$ est le jacobien du robot exprimé dans le repère R_n de son organe terminal [KHA 99];
- ${}^c\mathbb{T}_n$ est la matrice de transformation du torseur cinématique pour passer de son expression dans le repère de la caméra R_c au repère R_n . Cette matrice, constante si la caméra est rigidement liée à l'organe terminal du robot, est donnée par [KHA 99] :

$${}^c\mathbb{T}_n = \begin{pmatrix} {}^c\mathbf{R}_n & {}^c\tilde{\mathbf{P}}_n {}^c\mathbf{R}_n \\ \mathbf{0}_3 & {}^c\mathbf{R}_n \end{pmatrix} \quad [3.6]$$

où ${}^c\mathbf{R}_n$ et ${}^c\mathbf{P}_n$ sont respectivement la matrice de rotation et le vecteur de translation du repère R_c au repère R_n . Les éléments de la matrice de transformation du repère caméra au repère terminal peuvent être estimés assez précisément en utilisant une technique de calibration préalable, dite

bras-ciel [TSA 89, DOR 95, REM 98]. Les techniques d'asservissement visuel sont cependant en général assez robustes pour admettre des erreurs importantes de modélisation, aussi bien dans cette matrice de transformation [ESP 93, MAL 98] que dans le jacobien du robot.

Plus généralement, si une caméra embarquée observe un objet en mouvement, la différentielle de \mathbf{s} est alors donnée par :

$$\dot{\mathbf{s}} = \mathbf{L}_s {}^c\mathbb{T}_n {}^n\mathbf{J}_n(\mathbf{q}) \dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \quad [3.7]$$

où le terme $\frac{\partial \mathbf{s}}{\partial t}$ représente la variation de \mathbf{s} due au mouvement propre (et généralement inconnu) de l'objet. Dans le cas très improbable où le mouvement de l'objet est connu, et donné par exemple par le torseur cinématique \mathcal{T}_o dans R_c , on obtient finalement :

$$\dot{\mathbf{s}} = \mathbf{L}_s {}^c\mathbb{T}_n {}^n\mathbf{J}_n(\mathbf{q}) \dot{\mathbf{q}} - \mathbf{L}_s \mathcal{T}_o \quad [3.8]$$

3.2.3. Cas d'une caméra déportée

De manière similaire, si l'on considère à présent une caméra déportée observant l'organe terminal d'un robot manipulateur, la variation des informations visuelles liées rigidement à cet organe terminal s'écrit en fonction de la vitesse des coordonnées articulaires :

$$\dot{\mathbf{s}} = -\mathbf{L}_s {}^c\mathbb{T}_n {}^n\mathbf{J}_n(\mathbf{q}) \dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \quad [3.9]$$

où $\frac{\partial \mathbf{s}}{\partial t}$ décrit à présent les variations de \mathbf{s} dues à un éventuel mouvement de la caméra déportée.

REMARQUE.— On peut noter la différence de signe entre les équations [3.5] et [3.9]. Cette différence s'explique naturellement par le changement de configuration du capteur par rapport aux variables de contrôle (voir figure 3.2).

Que la caméra déportée soit fixe ou mobile, la matrice ${}^c\mathbb{T}_n$ est maintenant variable et doit être estimée à chaque itération, ce qui est fait généralement en utilisant une technique de localisation 3D (voir paragraphe 3.2.5.1). Si la caméra est fixe, il est donc préférable d'utiliser l'une des relations suivantes :

$$\dot{\mathbf{s}} = -\mathbf{L}_s {}^c\mathbb{T}_\emptyset {}^\emptyset\mathbb{T}_n {}^n\mathbf{J}_n(\mathbf{q}) \dot{\mathbf{q}} \quad [3.10]$$

$$= -\mathbf{L}_s {}^c\mathbb{T}_\emptyset \begin{pmatrix} \mathbb{I}_3 & {}^\emptyset\tilde{\mathbf{P}}_n \\ \mathbf{0}_3 & \mathbb{I}_3 \end{pmatrix} {}^\emptyset\mathbf{J}_n(\mathbf{q}) \dot{\mathbf{q}} \quad [3.11]$$

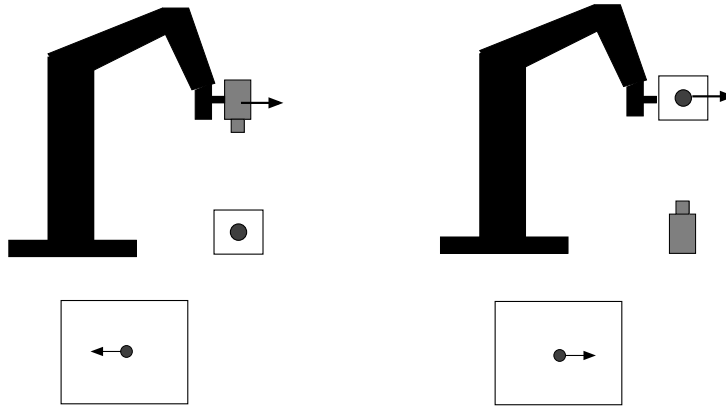


Figure 3.2. Différence de configuration (en haut) et d'effet produit dans l'image acquise par la caméra (en bas)

où ${}^0\mathbf{J}_n(\mathbf{q})$ est le jacobien du robot exprimé dans son repère de base et où les coefficients des matrices ${}^0\mathbb{T}_n$ et ${}^0\tilde{\mathbf{P}}_n$ sont fournis par le modèle géométrique direct du robot. En effet, la matrice de transformation ${}^c\mathbb{T}_\theta$ est alors constante et ne doit être estimée, généralement grossièrement, qu'une seule fois au préalable.

Dans la littérature [HAS 93a, HUT 96], la plupart des travaux se place dans le cadre d'un capteur embarqué. On peut cependant citer [ALL 93, NEL 94a, HAG 95, DOR 95, KEL 96, CIP 97, HOR 98, RUF 99] dans lesquels une ou plusieurs caméras déportées sont utilisées.

Dans tous les cas de figure, la matrice d'interaction joue un rôle essentiel et nous allons à présent donner sa forme analytique pour un ensemble d'informations visuelles de type géométrique. Par la suite, toutes les grandeurs nécessaires (coordonnées et vitesse de points, torseur cinématique, etc.) sont exprimées dans le repère de la caméra illustré sur la figure 3.3.

3.2.4. Matrice d'interaction d'informations 2D

3.2.4.1. Matrice d'interaction d'un point 2D

De manière très classique, le modèle mathématique d'une caméra est défini par une projection perspective telle que tout point M ayant pour coordonnées $\mathbf{M} = (X Y Z)^T$ se projette sur le plan image en un point m de coordonnées $(x y 1)^T$ avec :

$$x = X/Z \quad , \quad y = Y/Z \quad [3.12]$$

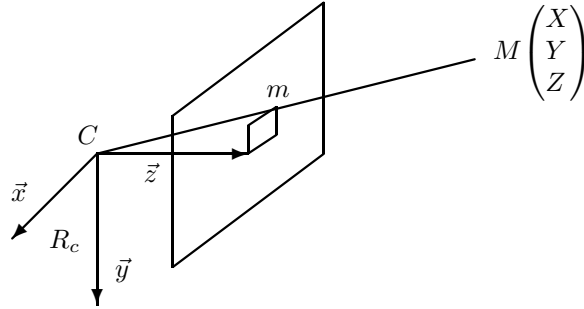


Figure 3.3. *Modèle de la caméra*

En différenciant cette équation, on obtient les variations dans l'image des coordonnées x et y de m par rapport à la vitesse $\dot{\mathbf{M}}$ des coordonnées du point M :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 1/Z & 0 & -X/Z^2 \\ 0 & 1/Z & -Y/Z^2 \end{pmatrix} \dot{\mathbf{M}} \quad [3.13]$$

Quelle que soit la configuration choisie (caméra embarquée ou déportée, point M fixe ou mobile), la vitesse $\dot{\mathbf{M}}$ de M en fonction du torseur cinématique \mathcal{T} entre la caméra et son environnement est donnée par l'équation fondamentale de la cinématique :

$$\dot{\mathbf{M}} = -\mathbf{V} - \boldsymbol{\omega} \times \mathbf{M} = -\mathbf{V} + \tilde{\mathbf{M}} \boldsymbol{\omega} = \begin{pmatrix} -\mathbb{I}_3 & \tilde{\mathbf{M}} \end{pmatrix} \mathcal{T} \quad [3.14]$$

L'équation [3.13] peut alors se simplifier, en utilisant l'équation [3.12], sous la forme :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \mathbf{L}_{xy}(x, y, Z) \mathcal{T} \quad [3.15]$$

où :

$$\mathbf{L}_{xy}(x, y, Z) = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \quad [3.16]$$

On peut noter que les termes induits par les mouvements de rotation ne dépendent que des mesures x et y dans l'image. Par contre, les termes induits par les mouvements de translation sont inversement proportionnels à la profondeur du point 3D considéré. Ce phénomène se retrouve pour toutes les informations visuelles qu'il est possible de définir dans l'image (et décrit l'ambiguïté classique en vision par ordinateur entre l'amplitude d'un mouvement de translation et

la profondeur des objets). En asservissement visuel, il faudra donc injecter une information 3D, pourtant a priori inconnue, dès que l'on souhaitera contrôler des degrés de liberté du robot impliquant des mouvements de translation.

Les algorithmes de traitement d'images fournissent des données exprimées en pixels. Si l'on néglige les phénomènes fortement non linéaires de distorsion dus par exemple à l'emploi d'objectifs à très courte focale, le changement de variable pour passer des coordonnées $(x_p y_p)^T$ d'un point, exprimées en pixels, aux coordonnées $(x y)^T$ de ce même point, mais exprimées en mètre, est donné par :

$$x = (x_p - x_c)/f_x \quad , \quad y = (y_p - y_c)/f_y \quad [3.17]$$

où $(x_c y_c)^T$ représente les coordonnées dans l'image du point principal et où $f_x = f/l_x$ et $f_y = f/l_y$ sont les rapports entre la distance focale f de l'objectif et les dimensions l_x et l_y d'un pixel [HOR 95]. Ces paramètres, appelés paramètres intrinsèques de la caméra, peuvent être estimés au préalable lors d'une phase de calibration [BEY 92, ROB 93], mais, comme pour les éléments de la matrice bras-ciel, des approximations grossières sont en général suffisantes pour conserver la stabilité des systèmes d'asservissement visuel [ESP 93, MAL 98, MAL 99].

Il est aussi possible de calculer la matrice d'interaction associée aux coordonnées d'un point exprimées directement en pixels. A partir du changement de variable réciproque à [3.17], donné par :

$$x_p = x_c + f_x x \quad , \quad y_p = y_c + f_y y \quad [3.18]$$

on obtient immédiatement :

$$\mathbf{L}_{x_p y_p} = \begin{pmatrix} f_x & 0 \\ 0 & f_y \end{pmatrix} \mathbf{L}_{xy} \quad [3.19]$$

où l'ensemble des termes contenus dans \mathbf{L}_{xy} , excepté bien entendu la profondeur Z , peuvent s'exprimer en fonction des paramètres intrinsèques et des coordonnées $(x_p y_p)^T$ à l'aide de [3.17]. Si besoin, on peut procéder de même pour les informations visuelles qui seront vues par la suite et travailler avec des données exprimées en pixels. L'intérêt principal d'avoir une forme analytique de la matrice d'interaction qui dépend explicitement des paramètres intrinsèques est que l'on peut alors étudier la sensibilité des systèmes d'asservissement visuel aux erreurs d'estimation ou d'approximation sur ces paramètres.

On peut finalement signaler les travaux récents décrits dans [RUF 99] ayant abouti à la modélisation directe de la matrice jacobienne \mathbf{J}_s telle que $\dot{s} = \mathbf{J}_s \dot{q}$ dans le cas où s est constitué des coordonnées d'un point situé sur l'organe terminal d'un robot et observé par deux caméras déportées : $s = (x_g y_g x_d y_d)$. L'intérêt d'une telle approche est qu'il n'est alors plus nécessaire de connaître le jacobien, et donc le modèle géométrique, du robot utilisé.

Si l'on considère à présent une caméra équipée d'un zoom commandable, fournissant ainsi un degré de liberté supplémentaire au système, on obtient tout aussi simplement à partir de [3.18] :

$$\begin{pmatrix} \dot{x}_p \\ \dot{y}_p \end{pmatrix} = \mathbf{L}_{x_p y_p} \mathcal{T} + \begin{pmatrix} (x_p - x_c)/f \\ (y_p - y_c)/f \end{pmatrix} \dot{f} \quad [3.20]$$

Pour des raisons purement technologiques (la plupart des zooms étant commandable uniquement en position, et non en vitesse), peu de travaux ont utilisé cette fonctionnalité amenant pourtant une redondance très intéressante par rapport au mouvement de translation le long de l'axe optique. On peut citer cependant [ESP 95, HOS 95a].

3.2.4.2. Matrice d'interaction d'une primitive géométrique paramétrable 2D

On peut également calculer la matrice d'interaction associée à des informations visuelles construites à partir de primitives géométriques paramétrables [CHA 90, ESP 92]. Pour cela, il suffit de définir les équations représentant :

- la nature et la configuration de la primitive dans la scène :

$$\mathbf{h}(X, Y, Z, P_1, \dots, P_n) = 0 \quad [3.21]$$

- sa projection dans le plan image :

$$\mathbf{g}(x, y, p_1, \dots, p_m) = 0 \quad [3.22]$$

- le lien entre la primitive 3D et son image (appelée dans le cas d'une primitive volumétrique la surface des limbes, voir figure 3.4) :

$$1/Z = \mu(x, y, P_1, \dots, P_l) = 0, \quad [3.23]$$

A titre d'exemple, si une droite de l'espace est représentée par l'intersection des deux plans suivants :

$$\mathbf{h}(X, Y, Z, \mathbf{P}) = \begin{cases} h_1 = A_1 X + B_1 Y + C_1 Z + D_1 = 0 \\ h_2 = A_2 X + B_2 Y + C_2 Z = 0 \end{cases} \quad [3.24]$$

on obtient immédiatement en utilisant les équations de la projection perspective [3.12] :

- la fonction μ à partir de h_1 :

$$1/Z = \alpha x + \beta y + \gamma \quad [3.25]$$

avec $\alpha = -A_1/D_1, \beta = -B_1/D_1$ et $\gamma = -C_1/D_1$.

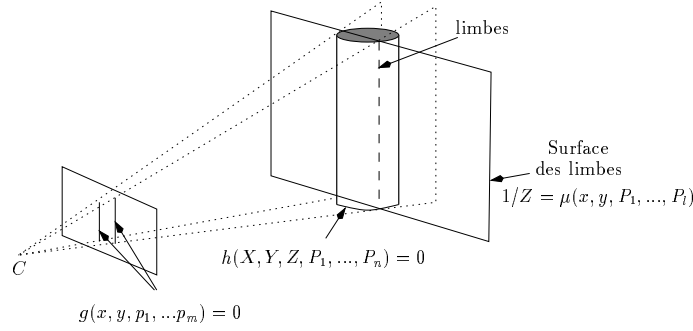


Figure 3.4. Projection de la primitive dans l'image et surface des limbes dans le cas d'un cylindre

- l'équation de la droite 2D, notée \mathcal{D} , résultant de la projection dans l'image de la droite 3D, à partir de h_2 :

$$ax + by + c = 0 \text{ avec } a = A_2, b = B_2, c = C_2 \quad [3.26]$$

La paramétrisation (a, b, c) n'étant pas minimale, on préfère choisir la représentation $\mathbf{p} = (\rho, \theta)$ définie par :

$$g(x, y, \mathbf{p}) = x \cos \theta + y \sin \theta - \rho = 0 \quad [3.27]$$

où $\theta = \arctan(b/a)$ et $\rho = -c/\sqrt{a^2 + b^2}$ (voir figure 3.5).

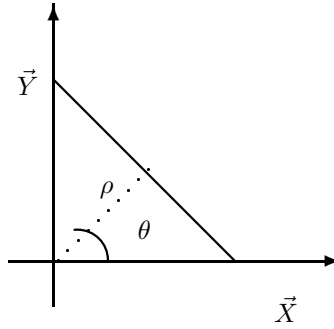


Figure 3.5. Représentation (ρ, θ) des droites 2D

En différenciant l'équation [3.27], ce qui correspond à l'hypothèse que l'image d'une droite restera une droite quels que soient les mouvements de la caméra, on obtient :

$$\dot{\rho} + (x \sin \theta - y \cos \theta) \dot{\theta} = \dot{x} \cos \theta + \dot{y} \sin \theta, \forall (x, y) \in \mathcal{D} \quad [3.28]$$

A partir de l'équation [3.27], on écrit x en fonction de y si $\cos \theta \neq 0$ (ou y en fonction de x sinon) et l'équation [3.28] peut alors s'écrire, en utilisant [3.15] et [3.25] :

$$(\dot{\rho} + \rho \tan \theta \dot{\theta}) + y (-\dot{\theta} / \cos \theta) = \mathbf{K}_1 \mathcal{T} + y \mathbf{K}_2 \mathcal{T}, \forall y \in \mathbb{R} \quad [3.29]$$

avec :

$$\begin{aligned} \mathbf{K}_1 &= \begin{pmatrix} \lambda_1 \cos \theta & \lambda_1 \sin \theta & -\lambda_1 \rho & \sin \theta & -\cos \theta - \rho^2 / \cos \theta & -\rho \tan \theta \end{pmatrix} \\ \mathbf{K}_2 &= \begin{pmatrix} \lambda_2 \cos \theta & \lambda_2 \sin \theta & -\lambda_2 \rho & \rho & \rho \tan \theta & 1 / \cos \theta \end{pmatrix} \end{aligned}$$

où $\lambda_1 = -\alpha \rho / \cos \theta - \gamma$ et $\lambda_2 = \alpha \tan \theta - \beta$.

On en déduit aussitôt :

$$\begin{cases} \dot{\rho} &= (\mathbf{K}_1 + \rho \sin \theta \mathbf{K}_2) \mathcal{T} \\ \dot{\theta} &= -\cos \theta \mathbf{K}_2 \mathcal{T} \end{cases} \quad [3.30]$$

d'où :

$$\begin{aligned} \mathbf{L}_\rho &= \begin{pmatrix} \lambda_\rho \cos \theta & \lambda_\rho \sin \theta & -\lambda_\rho \rho & (1 + \rho^2) \sin \theta & -(1 + \rho^2) \cos \theta & 0 \end{pmatrix} \\ \mathbf{L}_\theta &= \begin{pmatrix} \lambda_\theta \cos \theta & \lambda_\theta \sin \theta & -\lambda_\theta \rho & -\rho \cos \theta & -\rho \sin \theta & -1 \end{pmatrix} \end{aligned} \quad [3.31]$$

avec $\lambda_\rho = -\alpha \rho \cos \theta - \beta \rho \sin \theta - \gamma$ et $\lambda_\theta = -\alpha \sin \theta + \beta \cos \theta$.

On peut retrouver le même résultat en appliquant l'équation [3.28] à deux points de \mathcal{D} , par exemple ceux de coordonnées $(\rho \cos \theta \quad \rho \sin \theta)^T$ et $(\rho \cos \theta + \sin \theta \quad \rho \sin \theta - \cos \theta)^T$.

Les résultats pour des primitives plus complexes (cercles, sphères et cylindres) sont donnés en [CHA 90], ce qui permet donc d'utiliser en asservissement visuel des informations 2D associées à ces primitives. Il est de plus possible d'en déduire la matrice d'interaction associée à des informations définies à partir de plusieurs primitives (telles l'orientation d'un segment ou la distance d'un point à une droite par exemple). L'inconvénient est tout de même que l'on ne peut travailler que sur des environnements où de telles primitives géométriques existent (d'où l'emploi le plus fréquent de points caractéristiques).

3.2.4.3. Matrice d'interaction pour des formes 2D complexes

Des travaux récents ont permis d'établir la forme analytique de la matrice d'interaction associée à des informations visuelles représentant la projection dans l'image d'objets de forme complexe. Dans [COL 99b, DRU 99], les six termes correspondant à la partie affine de la transformation entre l'image d'un objet plan à sa position courante et l'image de ce même objet à sa position désirée sont considérés. Plus précisément, si $(x \ y)^T$ et $(x^* \ y^*)^T$ sont respectivement

les coordonnées dans l'image courante et dans l'image désirée d'un même point de l'objet, on suppose qu'il existe un jeu de paramètres $\Theta = (a_1 \ b_1 \ c_1 \ a_2 \ b_2 \ c_2)^T$ telle que la relation :

$$\begin{cases} x &= a_1 x^* + b_1 y^* + c_1 \\ y &= a_2 x^* + b_2 y^* + c_2 \end{cases} \quad [3.32]$$

soit valide pour tous les points de l'objet. Cette hypothèse n'est malheureusement pas vérifiée pour une caméra modélisée par une projection perspective. De plus, la matrice d'interaction associée à Θ présente une perte de rang (de 6 à 4) lorsque le plan de l'objet est parallèle au plan image.

Par ailleurs, si l'on décompose en série de Fourier la signature polaire $\rho(\theta)$ des points contours d'un objet dans l'image (définie telle que les coordonnées x et y d'un point du contour s'écrivent : $x = x_g + \rho(\theta) \cos \theta$, $y = y_g + \rho(\theta) \sin \theta$ où x_g et y_g sont les coordonnées du centre de gravité de l'objet), il est possible de calculer la matrice d'interaction associée aux termes de cette série [COL 99a]. La forme analytique obtenue est cependant peu sympathique.

On peut également calculer la variation des moments m_{ij} d'un objet, définis par :

$$m_{ij}(t) = \iint_{\mathcal{D}(t)} x^i y^j dx dy \quad [3.33]$$

où $\mathcal{D}(t)$ est le domaine occupé par l'objet dans l'image à l'instant t . Si l'on suppose que l'objet considéré est plan ou de surface des limbes plane et d'équation $1/Z = Ax + By + C$, on obtient pour l'aire $a (= m_{00})$ et les coordonnées $x_g (= m_{10}/m_{00})$ et $y_g (= m_{01}/m_{00})$ du centre de gravité de l'objet :

$$\begin{aligned} \mathbf{L}_a &= \begin{pmatrix} -aA & -aB & a(3/Z_g - C) & 3ay_g & -3ax_g & 0 \end{pmatrix} \\ \mathbf{L}_{x_g} &= \begin{pmatrix} -1/Z_g & 0 & x_g/Z_g + \epsilon_1 & x_g y_g + n_{11} & -(1 + x_g^2 + n_{20}) & y_g \end{pmatrix} \\ \mathbf{L}_{y_g} &= \begin{pmatrix} 0 & -1/Z_g & y_g/Z_g + \epsilon_2 & 1 + y_g^2 + n_{02} & -x_g y_g - n_{11} & -x_g \end{pmatrix} \end{aligned} \quad [3.34]$$

avec $1/Z_g = Ax_g + By_g + C$, $\epsilon_1 = An_{20} + Bn_{11}$, $\epsilon_2 = An_{11} + Bn_{02}$ et où n_{20} , n_{02} et n_{11} sont les moments centrés normalisés d'ordre 2 définis par :

$$n_{ij} = 4\mu_{ij}/a \text{ avec } \begin{cases} \mu_{20} = m_{20} - ax_g^2 \\ \mu_{02} = m_{02} - ay_g^2 \\ \mu_{11} = m_{11} - ax_g y_g \end{cases} \quad [3.35]$$

On note que la vitesse \dot{a} dépend linéairement de a , et est nulle pour tous les mouvements autres que ceux attendus de translation le long de l'axe optique de la caméra si l'objet est centré et parallèle au plan image ($A = B = x_g = y_g = 0$). On peut aussi remarquer que le résultat obtenu pour les coordonnées du centre de gravité d'un objet englobe celui donné en [3.15] pour un objet purement

ponctuel. En effet, pour un point, on a $n_{20} = n_{11} = n_{02} = 0$ et on peut poser $A = B = 0$ dans [3.34] pour retrouver exactement [3.15].

Si l'on s'intéresse aux moments centrés d'ordre 2, on a alors :

$$\begin{aligned} \mathbf{L}_{\mu_{20}} &= \begin{pmatrix} -3A\mu_{20} - 2B\mu_{11} & -B\mu_{20} & As_{20} + Bt_{20} + 4C\mu_{20} \\ t_{20} & -s_{20} & 2\mu_{11} \end{pmatrix} \\ \mathbf{L}_{\mu_{02}} &= \begin{pmatrix} -A\mu_{02} & -2A\mu_{11} - 3B\mu_{02} & As_{02} + Bt_{02} + 4C\mu_{02} \\ t_{02} & -s_{02} & -2\mu_{11} \end{pmatrix} \\ \mathbf{L}_{\mu_{11}} &= \begin{pmatrix} -2A\mu_{11} - B\mu_{02} & -2B\mu_{11} - A\mu_{20} & As_{11} + Bt_{11} + 4C\mu_{11} \\ t_{11} & -s_{11} & \mu_{02} - \mu_{20} \end{pmatrix} \end{aligned} \quad [3.36]$$

avec :

$$\begin{aligned} s_{20} &= 7x_g\mu_{20} + 5\mu_{30} & , & \quad t_{20} = 5(y_g\mu_{20} + \mu_{21}) + 2x_g\mu_{11} \\ s_{02} &= 5(x_g\mu_{02} + \mu_{12}) + 2y_g\mu_{11} & , & \quad t_{02} = 7y_g\mu_{02} + 5\mu_{03} \\ s_{11} &= 6x_g\mu_{11} + 5\mu_{21} + y_g\mu_{20} & , & \quad t_{11} = 6y_g\mu_{11} + 5\mu_{12} + x_g\mu_{02} \end{aligned}$$

où les moments centrés d'ordre 3 sont donnés par :

$$\begin{aligned} \mu_{30} &= m_{30} - 3x_gm_{20} + 2ax_g^3 & , & \quad \mu_{21} = m_{21} - 2x_gm_{11} - y_gm_{20} + 2ax_g^2y_g \\ \mu_{03} &= m_{03} - 3y_gm_{02} + 2ay_g^3 & , & \quad \mu_{12} = m_{12} - 2y_gm_{11} - x_gm_{02} + 2ax_gy_g^2 \end{aligned}$$

En choisissant pour \mathbf{s} les six moments d'ordre inférieur ou égal à 2 ($\mathbf{s} = (ax_g y_g \mu_{20} \mu_{11} \mu_{02})^T$), on peut espérer contrôler les six degrés de liberté d'un robot. Nous verrons dans la partie suivante que ce n'est malheureusement pas le cas.

3.2.4.4. Matrice d'interaction par apprentissage ou estimation

L'utilisation de la signature polaire ou des moments permet de considérer des objets de forme véritablement complexe, mais nécessite au niveau du traitement d'image une phase de segmentation spatiale qui peut s'avérer extrêmement délicate pour des environnements texturés. Pour éviter cette phase de segmentation et traiter des images quelconques, on peut opérer une analyse en composante principale de l'image désirée et en sélectionner les vecteurs propres principaux [NAY 96, DEG 97]. Les coefficients de cette décomposition constituent alors l'ensemble \mathbf{s} des informations visuelles. La forme analytique de la matrice d'interaction associée est alors inconnue (vue la difficulté à l'obtenir) et l'asservissement est basé sur une estimation purement numérique fournie par une technique d'apprentissage. Le principe de cette technique consiste à générer des mouvements sur les différents degrés de liberté disponibles et à mesurer la variation correspondante observée dans l'image.

Des techniques d'estimation de la matrice d'interaction ont également été utilisées sur des informations visuelles géométriques telles que celles décrites dans les paragraphes précédents. Elles sont toutes basées sur le même principe et s'effectuent soit hors ligne par apprentissage [WEI 84, RUF 99], éventuellement en employant des réseaux de neurones [SUH 93, WEL 96], soit en ligne au cours de l'asservissement [KIN 94, HOS 94, BOU 93, CHA 96, JAG 97]. Parmi ces travaux, on peut distinguer ceux qui sont basés sur une estimation purement numérique des termes de la matrice d'interaction [WEI 84, SUH 93, WEL 96, HOS 94, JAG 97], et ceux qui estiment les paramètres inconnus intervenant dans cette matrice, tels par exemple la structure des objets ou les paramètres intrinsèques de la caméra [KIN 94, CHA 96, RUF 99]. Le premier cas est très séduisant en pratique puisqu'il permet d'éviter toute phase de modélisation. L'inconvénient qui en résulte est qu'il est impossible de démontrer la stabilité du système en présence d'erreurs inévitables d'estimation. Le second cas de figure est donc plus satisfaisant sur le plan théorique, mais, comme il nécessite une détermination analytique préalable de la matrice d'interaction, il ne peut aujourd'hui s'appliquer à des schémas d'asservissement basés sur des informations visuelles aussi complexes que celles issues d'une analyse en composante principale de l'image.

3.2.5. *Matrice d'interaction d'informations 3D*

Comme cela a déjà été évoqué, il est également possible de choisir des informations visuelles exprimées non plus directement dans l'image, mais issues d'une phase de reconstruction ou de localisation 3D [WIL 96, MAR 97]. L'obtention de ces informations 3D s'effectue soit par simple triangulation si l'on dispose d'un système de vision stéréoscopique calibré, soit, dans le cas d'un capteur monoculaire, par vision dynamique ou par une méthode de calcul de pose [HOR 95]. Les techniques de vision dynamique reposent sur la mesure du mouvement de la caméra et du mouvement induit dans l'image. Elles sont en général assez sensibles aux erreurs de mesure [XIE 89, BOU 93, SMI 94, CHA 96]. Nous décrivons à présent brièvement les techniques de calcul de pose, car les plus couramment utilisées en asservissement visuel 3D.

3.2.5.1. *Calcul de pose*

De nombreuses méthodes existent pour estimer la situation d'une caméra par rapport à un objet en utilisant une image de cet objet. Elles reposent sur la connaissance a priori d'un modèle 3D de l'objet et des paramètres de calibration de la caméra. Plus précisément, pour une image acquise à l'instant t ,

elles fournissent une estimée $\hat{\mathbf{r}}(t)$ de la situation réelle $\mathbf{r}(t)$ entre le repère de la caméra et le repère de l'objet à partir des mesures $\mathbf{x}(t)$ extraites de l'image, des paramètres intrinsèques de la caméra et du modèle 3D de l'objet, représenté par exemple par l'ensemble \mathbf{X} des coordonnées 3D des points le constituant :

$$\hat{\mathbf{r}}(t) = \hat{\mathbf{r}}(\mathbf{x}(t), x_c, y_c, f_x, f_y, \mathbf{X}) \quad [3.37]$$

Les primitives géométriques à la base des mesures $\mathbf{x}(t)$ sont le plus souvent des points [HOR 89, HAR 89, DEM 95] ou des segments [LOW 87, DHO 89], voire des coniques [SAF 92, MA 93], ou encore des objets cylindriques [DHO 90]. Par contre, très peu de méthodes combinent des primitives de nature différente (voir cependant [PHO 95] pour l'utilisation conjointe de points et de droites).

Les méthodes décrites dans la littérature sont soit purement géométriques [HOR 89, DHO 89], soit numériques et de type linéaire itératif [DEM 95] ou purement non linéaire [LOW 87]. Sauf cas très particuliers [HOR 89], on ne dispose pas d'une solution analytique à ce problème inverse.

On peut signaler qu'en cas d'erreur dans les paramètres de calibration ou dans le modèle de l'objet, l'estimée $\hat{\mathbf{r}}(t)$ sera biaisée et, en raison de l'absence d'une solution analytique, il est malheureusement impossible de déterminer la valeur de ce biais. Il en est de même pour l'obtention de la matrice d'interaction associée à n'importe quelle information construite à partir de $\hat{\mathbf{r}}(t)$. On a en effet d'après [3.37] :

$$\dot{\hat{\mathbf{r}}}(t) = \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \mathcal{T} \quad [3.38]$$

d'où :

$$\mathbf{L}_{\hat{\mathbf{r}}} = \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \quad [3.39]$$

Le deuxième terme $\frac{\partial \mathbf{x}}{\partial \mathbf{r}}$ de ce produit de matrices n'est rien d'autre que la matrice d'interaction $\mathbf{L}_{\mathbf{x}}$ associée à \mathbf{x} . Il est donc connu si \mathbf{x} est constitué de primitives géométriques telles des points ou des segments de droite. Par contre, le premier terme $\frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}}$, qui représente la variation de l'estimation de $\hat{\mathbf{r}}$ en fonction d'une variation des informations visuelles, est inconnu. On peut juste noter qu'il est directement lié à la méthode d'estimation employée et dépend là encore des paramètres intrinsèques de la caméra et du modèle 3D de l'objet. C'est pourquoi nous supposons par la suite que l'estimation de $\hat{\mathbf{r}}(t)$ est parfaite, ce qui est le cas sous les (fortes) hypothèses que la caméra est parfaitement calibrée, que le modèle 3D de l'objet est parfaitement connu, que les mesures $\mathbf{x}(t)$ ne sont entachées d'aucune erreur, et que la méthode d'estimation ne souffre d'aucune instabilité numérique.

L'hypothèse la plus forte concerne la stabilité supposée de l'estimation aux erreurs de mesure. En effet, si l'on considère par exemple quatre points coplanaires, il n'existe théoriquement qu'une seule solution au problème de localisation [HOR 89], mais une très faible variation de la position des quatre points

dans l'image peut entraîner une très forte variation dans l'estimation de $\hat{\mathbf{r}}$ (la matrice $\left(\frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}}\right)$ est donc très mal conditionnée). Un tel phénomène est illustré sur la figure 3.6. En pratique, ce phénomène s'amenuise en considérant un nombre de points plus important, ou des points non coplanaires, mais nous ne disposons actuellement d'aucun résultat théorique sur la sensibilité des méthodes d'estimation et des mesures à choisir, quant à leur type, mais aussi quant à leur disposition dans l'image et dans l'espace 3D.

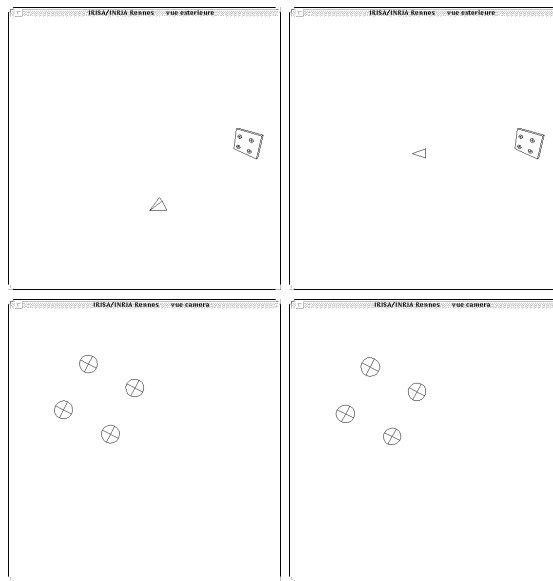


Figure 3.6. Exemple de deux positions distinctes de la caméra par rapport à l'objet (en haut) fournissant des images similaires de cet objet (en bas)

A partir de $\hat{\mathbf{r}}(t)$, et sous les hypothèses précitées telles que l'estimation de $\hat{\mathbf{r}}(t)$ est parfaite ($\hat{\mathbf{r}}(t) = \mathbf{r}(t)$), on dispose de la rotation ${}^c\mathbf{R}_o$ entre le repère de la caméra à sa position courante R_c et le repère R_o lié à l'objet, ainsi que de la translation ${}^c\mathbf{P}_o$ entre ces deux repères. On peut alors en déduire la position dans R_c de n'importe quel point lié à l'objet. Si, de plus, dans le cadre d'une caméra embarquée, la situation entre le repère de la caméra à sa position désirée R_{c^*} et le repère de l'objet est connue, on peut alors en déduire le déplacement à réaliser entre R_c et R_{c^*} . Avec une caméra déportée fixe, il en est bien sûr de même pour le déplacement de l'objet lié à l'organe terminal du robot entre sa position courante et sa position désirée.

Nous donnons à présent la matrice d'interaction associée à la représentation minimale $\theta \mathbf{u}$ d'une rotation quelconque d'angle θ autour d'un axe \mathbf{u} , puis celle associée aux coordonnées d'un point 3D.

3.2.5.2. Matrice d'interaction associée à $\theta\mathbf{u}$

On rappelle tout d'abord que la représentation $\theta\mathbf{u}$ s'obtient de manière unique à partir des coefficients $r_{ij(i=1..3,j=1..3)}$ d'une matrice de rotation \mathbf{R} à l'aide de l'équation suivante [KHA 99, MAL 98] :

$$\theta\mathbf{u} = \frac{1}{2 \operatorname{sinc}\theta} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix} \quad [3.40]$$

où $\theta = \arccos((r_{11} + r_{22} + r_{33} - 1)/2)$ et où le sinus cardinal $\operatorname{sinc}\theta$, défini par $\sin\theta = \theta \operatorname{sinc}\theta$, est une fonction C^∞ s'annulant en $(2n + 1)\pi, \forall n \in \mathbb{Z}$. Pour $\theta = \pi$, seul cas non pris en compte par [3.40], \mathbf{u} est le vecteur propre de \mathbf{R} associé à la valeur propre 1.

La matrice d'interaction associée à $\theta\mathbf{u}$ est donnée par [MAL 98] :

$$\mathbf{L}_{\theta\mathbf{u}} = \begin{pmatrix} \mathbf{0}_3 & \mathbf{L}_\omega \end{pmatrix} \quad [3.41]$$

avec :

$$\mathbf{L}_\omega = \mathbb{I}_3 - \frac{\theta}{2} \tilde{\mathbf{u}} + \left(1 - \frac{\operatorname{sinc}\theta}{\operatorname{sinc}^2\frac{\theta}{2}}\right) \tilde{\mathbf{u}}^2 \quad [3.42]$$

La représentation $\theta\mathbf{u}$ est donc particulièrement intéressante puisque \mathbf{L}_ω est singulière uniquement pour $\theta = 2\pi$. Par ailleurs, on a :

$$\mathbf{L}_\omega^{-1} = \mathbb{I}_3 + \frac{\theta}{2} \operatorname{sinc}^2\frac{\theta}{2} \tilde{\mathbf{u}} + (1 - \operatorname{sinc}\theta) \tilde{\mathbf{u}}^2 \quad [3.43]$$

qui nous assure la propriété bien pratique suivante :

$$\mathbf{L}_\omega^{-1} \theta\mathbf{u} = \theta\mathbf{u} \quad [3.44]$$

Si l'on dispose d'une caméra embarquée, on peut utiliser en asservissement visuel le vecteur $\theta\mathbf{u}$ pour représenter la rotation ${}^{c^*}\mathbf{R}_c$ entre R_{c^*} et R_c . Si les matrices ${}^{c^*}\mathbf{R}_{n^*}$ et ${}^c\mathbf{R}_n$ sont identiques, ce qui est généralement le cas, on peut également considérer le vecteur $\theta\mathbf{u}$ associée à la rotation ${}^{n^*}\mathbf{R}_n$. De même, avec une caméra déportée, on peut utiliser le vecteur $\theta\mathbf{u}$ pour représenter soit la rotation ${}^{o^*}\mathbf{R}_o$ entre les repères désiré et courant de l'objet lié à l'effecteur, soit la rotation ${}^{n^*}\mathbf{R}_n$ si les matrices ${}^{o^*}\mathbf{R}_{n^*}$ et ${}^o\mathbf{R}_n$ sont identiques (ce qui est également généralement le cas).

On peut noter qu'il ne faut pas prendre en compte directement le vecteur $\theta\mathbf{u}$ associé à la rotation ${}^c\mathbf{R}_o$ et raisonner sur la différence entre $\theta\mathbf{u}$ et $\theta^*\mathbf{u}^*$ (où $\theta^*\mathbf{u}^*$ représente la rotation à atteindre ${}^{c^*}\mathbf{R}_o$). En effet, $\theta\mathbf{u} - \theta^*\mathbf{u}^*$ ne représente pas une distance dans l'espace SO_3 des rotations [SAM 91].

3.2.5.3. Matrice d'interaction associée à un point 3D

A partir de l'équation fondamentale de la cinématique rappelée en [3.14], on obtient immédiatement pour tout point de coordonnées \mathbf{X} lié à l'objet :

$$\mathbf{L}_{\mathbf{X}} = \begin{pmatrix} -\mathbb{I}_3 & \tilde{\mathbf{X}} \end{pmatrix} \quad [3.45]$$

Les points pris en compte peuvent être des points caractéristiques de l'objet [MAR 96b], ou encore l'origine de R_o (on a alors $\mathbf{X} = {}^c\mathbf{P}_o$).

Avec une caméra embarquée, si l'on s'intéresse au déplacement qu'elle doit réaliser, on peut aussi considérer l'origine de R_{c^*} (on a alors $\mathbf{X} = {}^c\mathbf{P}_{c^*}$ et $\mathbf{X}^* = \mathbf{0}$) [MAR 97]. Dans ce cas, il est encore plus intéressant de considérer la position de l'origine du repère de la caméra exprimée dans un repère lié à l'objet, tel R_o , R_{c^*} , voire R_\emptyset si l'objet est fixe (voir figure 3.7) [WIL 96].

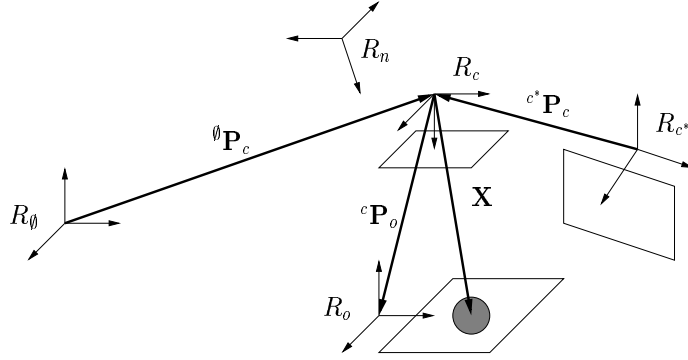


Figure 3.7. Points 3D possibles avec une caméra embarquée

Si l'on choisit par exemple R_o , on a :

$${}^o\mathbf{P}_c = -{}^c\mathbf{R}_o^T {}^c\mathbf{P}_o = -{}^o\mathbf{R}_c {}^c\mathbf{P}_o \quad [3.46]$$

En dérivant cette équation, on obtient :

$$\begin{aligned} {}^o\dot{\mathbf{P}}_c &= -{}^o\dot{\mathbf{R}}_c {}^c\mathbf{P}_o - {}^o\mathbf{R}_c {}^c\dot{\mathbf{P}}_o \\ &= -{}^o\mathbf{R}_c ({}^c\mathbf{R}_o {}^o\dot{\mathbf{R}}_c {}^c\mathbf{P}_o + {}^c\dot{\mathbf{P}}_o) \end{aligned}$$

soit, en utilisant [3.4] et [3.45] :

$$\begin{aligned} {}^o\dot{\mathbf{P}}_c &= -{}^o\mathbf{R}_c (\tilde{\omega} {}^c\mathbf{P}_o - \mathbf{V} + {}^c\tilde{\mathbf{P}}_o \omega) \\ &= {}^o\mathbf{R}_c \mathbf{V} \end{aligned}$$

On a donc :

$$\mathbf{L}_{{}^o\mathbf{P}_c} = \begin{pmatrix} {}^o\mathbf{R}_c & \mathbf{0}_3 \end{pmatrix} \quad [3.47]$$

qui est indépendant des mouvements de rotation de la caméra. De même, en choisissant ${}^c\mathbf{P}_c$, on obtient :

$$\mathbf{L}_{c^*\mathbf{P}_c} = \begin{pmatrix} {}^c\mathbf{R}_c & \mathbf{0}_3 \end{pmatrix} \quad [3.48]$$

et on aura alors ${}^c\mathbf{P}_c^* = \mathbf{0}$.

Avec une caméra déportée (voir figure 3.8), et pour les mêmes propriétés de découplage, il est intéressant de considérer la position de l'origine des repères R_o ou R_n , et d'exprimer le torseur cinématique en cette origine. En effet, si l'on choisit par exemple ${}^c\mathbf{P}_o$, on a alors en utilisant [3.45] et [3.6] :

$$\mathbf{L}_{c\mathbf{P}_o} {}^c\mathcal{T}_o = \begin{pmatrix} -\mathbb{I}_3 & {}^c\tilde{\mathbf{P}}_o \end{pmatrix} \begin{pmatrix} {}^c\mathbf{R}_o & {}^c\tilde{\mathbf{P}}_o {}^c\mathbf{R}_o \\ \mathbf{0}_3 & {}^c\mathbf{R}_o \end{pmatrix} \quad [3.49]$$

soit :

$$\mathbf{L}_{c\mathbf{P}_o} {}^c\mathcal{T}_o = \begin{pmatrix} -{}^c\mathbf{R}_o & \mathbf{0}_3 \end{pmatrix} \quad [3.50]$$

On peut bien sûr exprimer la position de l'origine de R_o dans n'importe quel repère. Si l'on retient le repère de base du robot R_θ , on obtient simplement :

$${}^\theta\dot{\mathbf{P}}_o = \begin{pmatrix} \mathbb{I}_3 & \mathbf{0}_3 \end{pmatrix} {}^\theta\mathcal{T}_o \quad [3.51]$$

où ${}^\theta\mathcal{T}_o$ est le torseur cinématique de l'objet exprimé dans R_θ et en l'origine de R_o . Le même résultat est bien sûr atteint si l'on considère ${}^\theta\mathbf{P}_n$ et ${}^\theta\mathcal{T}_n$.

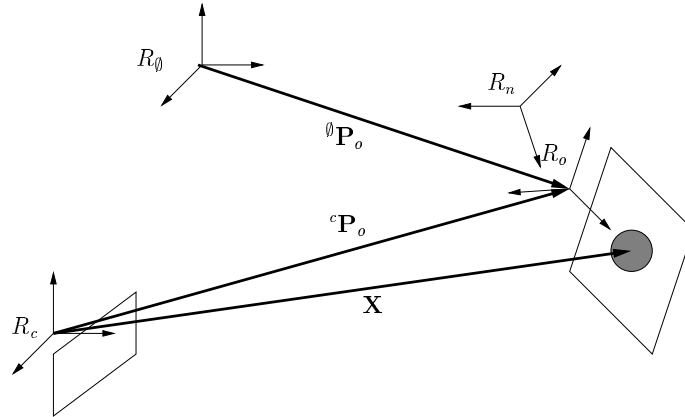


Figure 3.8. Points 3D possibles avec une caméra déportée

3.2.5.4. Matrice d'interaction associée à un plan 3D

Finalement, on peut aussi déterminer la matrice d'interaction associée à des primitives géométriques 3D telles que des segments, des plans, des sphères, etc.

Par exemple, dans le cas d'un plan représenté par sa normale unitaire \mathbf{U} et sa distance à l'origine D , on obtient [MOT 92] :

$$\mathbf{L}_{(\mathbf{U},D)} = \begin{pmatrix} \mathbf{0}_3 & \tilde{\mathbf{U}} \\ \mathbf{U}^T & \mathbf{0} \end{pmatrix} \quad [3.52]$$

3.3. Fonction de tâche et commande

La réalisation d'une tâche robotique par asservissement visuel nécessite la sélection d'informations visuelles adéquates et l'élaboration d'une loi de commande en boucle fermée sur ces informations. La première phase revient à définir une fonction de tâche ayant des propriétés assurant la capacité de réaliser la tâche choisie [SAM 91], la seconde à réguler cette fonction de tâche. Nous considérons tout d'abord le cas où l'on souhaite contrôler les 6 degrés de liberté du système, c'est-à-dire amener le repère lié à l'effecteur du robot à une situation unique désirée.

Si l'on utilise un ensemble de k informations visuelles \mathbf{s} , la fonction de tâche \mathbf{e} s'écrit de manière générale :

$$\mathbf{e}(\mathbf{r}(t)) = \mathbf{C} (\mathbf{s}(\mathbf{r}(t)) - \mathbf{s}^*) \quad [3.53]$$

où :

- $\mathbf{s}(\mathbf{r}(t))$ est la valeur courante des informations visuelles sélectionnées ;
- \mathbf{s}^* est la valeur que \mathbf{s} doit atteindre pour réaliser la tâche ;
- \mathbf{C} est une matrice $6 \times k$ de rang plein 6, dite de combinaison, telles que les 6 composantes de \mathbf{e} soient indépendantes et contrôlent les 6 degrés de liberté du système.

3.3.1. Obtention de la consigne \mathbf{s}^*

Quelle que soit la nature des informations visuelles choisies, la consigne \mathbf{s}^* s'obtient très généralement soit en fixant a priori la situation à atteindre entre la caméra et l'objet d'intérêt, soit par apprentissage.

- Dans le premier cas, si \mathbf{s} comprend des informations 2D, leur valeur désirée s'obtient aisément si un modèle de l'objet est disponible : il suffit alors d'appliquer les équations de projection perspective pour calculer la position de l'objet dans l'image. De plus, il est aussi possible de spécifier la situation à atteindre entre l'effecteur du robot et l'objet d'intérêt (pour une tâche de préhension par exemple) : le calcul des informations visuelles (2D ou 3D) s'obtient alors immédiatement si la matrice de passage entre

l'effecteur et la caméra est connue. Cependant, dans tous les cas de figure, toute erreur de modélisation sur les paramètres de calibration de la caméra, sur le modèle de l'objet (et éventuellement sur la matrice de passage effecteur-caméra) aura pour conséquence que, quand la valeur de \mathbf{s} sera égale à \mathbf{s}^* , la situation réellement atteinte sera différente de celle spécifiée, en raison du biais introduit par les erreurs de modélisation.

- L'obtention de la consigne \mathbf{s}^* par apprentissage, si elle est plus contraignante à mettre en œuvre en pratique, est donc préférable pour assurer une bonne réalisation de la tâche. Il s'agit d'amener dans une phase préalable le robot à sa position désirée par rapport à l'objet, d'acquérir l'image correspondante, et de calculer la valeur de \mathbf{s}^* exactement de la même manière que pour les futurs calculs de $\mathbf{s}(t)$. En présence d'erreur de modélisation, on se retrouve alors dans la situation paradoxale où la consigne et les mesures sont biaisées, mais où la situation atteinte après convergence est correcte aux seules erreurs de mesure près.
- Une troisième solution, plus élégante, consiste à faire en sorte que la caméra observe simultanément l'effecteur du robot et l'objet d'intérêt. Le calcul de \mathbf{s}^* peut alors s'effectuer automatiquement [HOR 98]. Cette solution n'a été que rarement appliquée, car, si elle semble naturelle avec des caméras déportées, elle pose de difficiles problèmes de placement de capteurs dans le cas des systèmes embarqués.

Nous détaillons à présent les différents choix possibles pour la matrice de combinaison à travers une analyse (simple) de la stabilité du système.

3.3.2. Régulation de la fonction de tâche

Comme nous l'avons vu au début de cette section, l'élaboration de la loi de commande pour réguler la fonction de tâche peut être dissociée de la définition de cette fonction. Dans la littérature, de nombreux types de lois de commande ont été proposés : loi de commande non linéaire [HAS 93b, REY 98], optimale de type LQ ou LQG [PAP 93, HAS 96], basée sur un contrôleur GPC [GAN 98, GAN 99], voire robuste H_∞ [KHA 96b] ou par retour d'état continu instationnaire [TSA 98] dans le cas de robots mobiles présentant des contraintes non-holonomes. On va simplement s'intéresser ici à réaliser une décroissance exponentielle de la fonction de tâche, à savoir :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad [3.54]$$

En utilisant [3.53] et [3.2], si la matrice \mathbf{C} est choisie constante, la différentielle de \mathbf{e} est donnée par :

$$\dot{\mathbf{e}} = \mathbf{C} \dot{\mathbf{s}} = \mathbf{C} \mathbf{L}_s \mathcal{T} \quad [3.55]$$

On a vu aux paragraphes 3.2.2 et 3.2.3 comment passer du torseur cinématique \mathcal{T} aux variables articulaires $\dot{\mathbf{q}}$. Nous supposons donc par la suite pour simplifier les notations que la grandeur de commande est simplement la partie commandable de \mathcal{T} , notée $\mathcal{T}_{\mathbf{q}}$ (on ne considère donc pas ici les problèmes dus aux singularités et aux butées du robot et on a bien sûr $\mathcal{T}_{\mathbf{q}} = \mathcal{T}_{\mathbf{c}}$ dans le cas d'une caméra embarquée et $\mathcal{T}_{\mathbf{q}} = -\mathcal{T}_{\mathbf{o}}$ dans le cas d'une caméra déportée). On écrit alors :

$$\dot{\mathbf{e}} = \mathbf{C} \mathbf{L}_{\mathbf{s}} \mathcal{T}_{\mathbf{q}} + \frac{\partial \mathbf{e}}{\partial t} \quad [3.56]$$

où $\frac{\partial \mathbf{e}}{\partial t}$ représente les variations de \mathbf{e} dues soit au mouvement de l'objet (si l'on dispose d'une caméra embarquée), soit au mouvement de la caméra (si celle-ci est déportée et mobile). Pour contrôler les 6 degrés de liberté du système, il faut pour le moins sélectionner \mathbf{s} tel que $\mathbf{L}_{\mathbf{s}}$ soit de rang 6 et on obtient comme loi de commande idéale :

$$\mathcal{T}_{\mathbf{q}} = (\mathbf{C} \mathbf{L}_{\mathbf{s}})^{-1} \left(-\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t} \right) \quad [3.57]$$

Dans le cas où les informations visuelles sont exprimées dans l'image, on a vu que la matrice d'interaction dépend de la valeur de ces informations visuelles et de la profondeur entre la caméra et l'objet d'intérêt. Dans le cas où les informations visuelles sont de type 3D, seules des hypothèses assez fortes permettent l'obtention analytique de cette matrice. Dans tous les cas de figure, des erreurs de mesure et d'estimation sont inévitables et la valeur exacte de $\mathbf{L}_{\mathbf{s}}$ est inconnue. Seule une approximation $\widehat{\mathbf{L}}_{\mathbf{s}}$ peut donc être considérée dans la loi de commande. De plus, le terme $\frac{\partial \mathbf{e}}{\partial t}$ est en général inconnu. La loi de commande utilisée en pratique est donc :

$$\mathcal{T}_{\mathbf{q}} = (\mathbf{C} \widehat{\mathbf{L}}_{\mathbf{s}})^{-1} \left(-\lambda \mathbf{e} - \frac{\partial \widehat{\mathbf{e}}}{\partial t} \right) \quad [3.58]$$

Si on suppose que cette consigne est parfaitement réalisée, l'utilisation de [3.58] dans [3.56] donne :

$$\dot{\mathbf{e}} = -\lambda \mathbf{C} \mathbf{L}_{\mathbf{s}} (\mathbf{C} \widehat{\mathbf{L}}_{\mathbf{s}})^{-1} \mathbf{e} - \mathbf{C} \mathbf{L}_{\mathbf{s}} (\mathbf{C} \widehat{\mathbf{L}}_{\mathbf{s}})^{-1} \frac{\partial \widehat{\mathbf{e}}}{\partial t} + \frac{\partial \mathbf{e}}{\partial t} \quad [3.59]$$

On constate alors, en supposant pour simplifier que $\frac{\partial \mathbf{e}}{\partial t} = \frac{\partial \widehat{\mathbf{e}}}{\partial t} = 0$, que la condition de positivité :

$$\mathbf{C} \mathbf{L}_{\mathbf{s}} (\mathbf{C} \widehat{\mathbf{L}}_{\mathbf{s}})^{-1} > 0 \quad [3.60]$$

est suffisante à assurer la décroissance de $\|\mathbf{e}\|$. De plus, on retrouvera le comportement spécifié en [3.54] sous l'unique condition que $\widehat{\mathbf{L}}_{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}$ et que $\frac{\partial \widehat{\mathbf{e}}}{\partial t} = \frac{\partial \mathbf{e}}{\partial t}$.

Nous verrons au paragraphe 3.3.4 comment il est possible d'estimer $\frac{\partial \mathbf{e}}{\partial t}$, ce qui permet alors de réduire autant que possible les erreurs de traînage. Nous nous intéressons à présent aux différents choix possibles pour \mathbf{C} et $\widehat{\mathbf{L}}_s$. Nous supposons donc par la suite que $\frac{\partial \mathbf{e}}{\partial t} = \frac{\partial \widehat{\mathbf{e}}}{\partial t} = 0$ pour ne pas alourdir les notations.

3.3.2.1. Cas où \mathbf{s} est de dimension 6 ($k = 6$)

Si \mathbf{s} est de dimension 6, on a tout intérêt à choisir $\mathbf{C} = \mathbb{I}_6$. En effet, le comportement de \mathbf{s} sera alors le même que celui de \mathbf{e} (c'est-à-dire que dans le cas idéal, toutes les composantes de \mathbf{s} décroîtront de manière exponentielle découplée). Dans ce cas, on obtient comme loi de commande :

$$\mathcal{T}_{\mathbf{q}} = -\lambda \widehat{\mathbf{L}}_s^{-1} \mathbf{e} = -\lambda \widehat{\mathbf{L}}_s^{-1} (\mathbf{s} - \mathbf{s}^*) \quad [3.61]$$

et comme condition de stabilité :

$$\mathbf{L}_s \widehat{\mathbf{L}}_s^{-1} > 0 \quad [3.62]$$

Si l'on est en mesure d'estimer correctement la valeur courante de \mathbf{L}_s à chaque itération de la loi de commande, la prise en compte de cette estimation permettra de se rapprocher au mieux du comportement idéal $\dot{\mathbf{s}} = -\lambda \mathbf{s}$.

3.3.2.1.1. Informations visuelles 2D

En considérant des informations visuelles 2D, il est malheureusement impossible (à ce jour) de se retrouver dans ce cas de figure sans limiter fortement l'espace de travail. La difficulté principale ne porte pas sur l'estimation de la valeur courante de la matrice d'interaction, mais sur la sélection des six informations visuelles.

Considérons par exemple le cas où \mathbf{s} est constitué des coordonnées de la projection de trois points dans l'image. La matrice d'interaction associée \mathbf{L}_s est alors de dimension 6×6 et est, dans la plupart des cas, de rang plein 6. Mais, on peut montrer [MIC 93, PAP 95] que certaines configurations entraînent une perte de rang de \mathbf{L}_s . En l'occurrence, ces singularités sont telles que les trois points soient alignés dans l'image ou que le centre optique C de la caméra appartienne à la surface du cylindre défini par le cercle circonscrit aux trois points (voir figure 3.9). Il est donc difficile d'assurer que, quelle que soit la position initiale choisie, le mouvement du robot évitera de traverser une singularité isolée (où évidemment la condition de stabilité [3.62] n'est plus respectée).

De plus, il existe généralement quatre situations distinctes entre la caméra et la scène telles que l'image de trois points soit la même [DHO 89]. La minimisation de $\|\mathbf{s} - \mathbf{s}^*\|$ peut donc amener le robot à l'un des quatre minima globaux

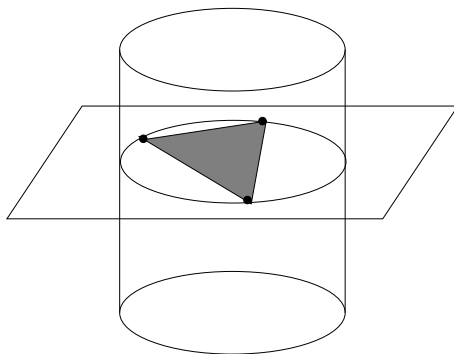


Figure 3.9. *Cylindre de singularité*

tels que $\|\mathbf{s} - \mathbf{s}^*\| = 0$. Ainsi, dans ce cas, on peut très bien avoir $\mathbf{s} - \mathbf{s}^* = 0$ sans que la situation atteinte soit celle qui a été spécifiée.

En considérant des informations visuelles de nature différente (telles par exemple les 3 droites que l'on peut définir à partir de 3 points non alignés), on retombe bien évidemment sur les mêmes problèmes potentiels. Pour y remédier, une solution pratique consiste à restreindre l'espace de travail à des zones proches de la situation désirée, ne comportant pas de singularités isolées, et où la minimisation de $\|\mathbf{s} - \mathbf{s}^*\|$ attire l'effecteur du robot vers sa position désirée. La détermination de la taille de ces zones est cependant très problématique.

Par ailleurs, si l'on utilise les 6 premiers moments d'un objet dans l'image, on peut montrer facilement à partir des équations données en 3.2.4.3 que la matrice d'interaction associée est au plus de rang 5 si la forme de l'objet dans l'image est telle que ses moments centrés d'ordre 3 sont tous nuls (ce qui est le cas pour un rectangle par exemple). En considérant des objets de forme plus complexe, la matrice d'interaction est alors de rang 6, mais son conditionnement est médiocre et ne permet pas d'obtenir des résultats satisfaisants. De même, si l'on utilise les six termes décrivant la déformation affine d'un objet (voir 3.2.4.3), une perte de rang de la matrice d'interaction intervient lorsque l'objet considéré est parallèle au plan image [COL 99b, DRU 99].

Pour les différentes raisons évoquées ci-dessus, on utilise très généralement des informations visuelles 2D redondantes. On est alors dans le cas où k est supérieur à 6, qui sera décrit au paragraphe 3.3.2.2.

3.3.2.1.2. Informations visuelles 3D

L'emploi d'informations visuelles 3D permet d'éviter les problèmes évoqués ci-dessus puisque l'on dispose alors des trois paramètres $\theta_{\mathbf{u}}$ pour représenter

l'orientation et qu'il suffit de choisir trois paramètres de position parmi ceux donnés en 3.2.5.3 pour avoir $k = 6$. On rappelle tout de même qu'il est alors nécessaire de se placer dans le cas idéal où les diverses erreurs de mesure, de calibration et d'estimation sont négligeables pour pouvoir exprimer la matrice d'interaction. Dans la suite de ce paragraphe, nous nous plaçons donc dans ce cas idéal qui assure (en théorie) le comportement spécifié $\dot{\mathbf{s}} = -\lambda \mathbf{s}$ et la condition de stabilité [3.62] dans tout l'espace de travail. On a alors en effet :

$$\mathbf{L}_s \widehat{\mathbf{L}}_s^{-1} = \mathbf{L}_s \mathbf{L}_s^{-1} = \mathbb{I}_6 > 0 \quad [3.63]$$

Si l'on choisit d'utiliser les coordonnées \mathbf{X} d'un point lié à l'objet exprimé dans le repère courant de la caméra R_c , la matrice d'interaction globale associée à $\mathbf{s} = (\mathbf{X} \ \theta \mathbf{u})^T$ est donnée par :

$$\mathbf{L}_s = \begin{pmatrix} -\mathbb{I}_3 & \tilde{\mathbf{X}} \\ \mathbf{0}_3 & \mathbf{L}_\omega \end{pmatrix} \quad [3.64]$$

On peut noter la forme sympathique de cette matrice (bloc triangulaire et non singulière sauf en $\theta = 2\pi$), qui va donner un comportement intéressant au système. En effet, idéalement, la trajectoire du point considéré sera rectiligne dans l'image [MAL 98]. Si ce point est dans le champ de vision de la caméra en ses positions initiale et désirée, il le sera donc en permanence. En sélectionnant correctement ce point (au centre de gravité de l'objet par exemple), on peut alors minimiser le risque de perdre une grande partie de l'objet d'intérêt au cours de l'asservissement (sans toutefois assurer que suffisamment de caractéristiques nécessaires au calcul de pose restent visibles). À chaque itération, on peut également sélectionner le point 3D correspondant au point 2D le plus proche des limites du plan image. Ce choix impliquera cependant une discontinuité dans les composantes en translation de la commande à chaque changement de point. De plus, sans stratégie de plus haut niveau (telle une planification de trajectoires dans l'image), il est toujours possible d'exhiber des cas particuliers qui vont soit entraîner la perte de vue d'une partie de l'objet, soit induire des effets pervers sur la commande (si deux points sont proches de bords opposés de l'image par exemple). Finalement, la trajectoire en translation qu'effectuera la caméra sera rectiligne, mais seulement dans le repère **mobile** de la caméra. Elle ne sera donc pas une véritable ligne droite si un changement d'orientation est nécessaire.

Avec une caméra embarquée, on obtient également une matrice d'interaction bloc-triangulaire si l'on considère le vecteur $\mathbf{X} = {}^c\mathbf{P}_{c^*}$ pour contrôler la position de la caméra. Ce choix est cependant peu intéressant en pratique. Par contre, si l'on choisit de considérer la position de l'origine de R_c exprimée dans un repère

lié à l'objet, soit par exemple ${}^c\mathbf{P}_c$, on a alors (voir [3.48]) :

$$\mathbf{L}_s = \begin{pmatrix} {}^c\mathbf{R}_c & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{L}_\omega \end{pmatrix} \quad [3.65]$$

qui est bloc-diagonale et assure donc un découplage très satisfaisant. De plus, la trajectoire en translation de la caméra sera alors une véritable ligne droite, ce qui est d'un grand intérêt pratique. Si l'on souhaite que ce soit l'effecteur du robot qui suive une ligne droite, il suffit de considérer ${}^n\mathbf{P}_n$ dans \mathbf{s} à la place de ${}^c\mathbf{P}_c$. Malheureusement, dans les deux cas, il n'y alors plus aucun contrôle de la trajectoire qu'effectuera l'objet dans l'image et, si la position initiale de la caméra est éloignée de sa position désirée, rien n'assure que l'objet restera dans le champ de vision de la caméra au cours de l'asservissement.

On peut bien entendu effectuer des choix similaires avec une caméra déportée (voir paragraphe 3.2.5.3). A titre d'exemple, si l'on sélectionne dans \mathbf{s} la translation ${}^\theta\mathbf{P}_o$ et le vecteur $\theta\mathbf{u}$ associé à la rotation ${}^o\mathbf{R}_o$, on aboutit, en combinant les équations [3.61], [3.51] et [3.44], à la loi de commande suivante :

$$\begin{cases} {}^\theta\mathbf{V}_o = -\lambda ({}^\theta\mathbf{P}_o - {}^\theta\mathbf{P}_{o^*}) \\ {}^o\omega_o = -\lambda \theta \mathbf{u} \end{cases} \quad [3.66]$$

où ${}^\theta\mathbf{V}_o$ est la vitesse de translation de R_o exprimée dans R_θ et où ${}^o\omega_o$ est sa vitesse de rotation exprimée dans R_o . Cette loi de commande présente des propriétés de découplage idéales. Elle assure en outre que la trajectoire de l'origine de R_o sera rectiligne, aussi bien dans l'espace 3D que dans l'image. Il faut toutefois signaler qu'en présence d'erreur de modélisation dans le jacobien du robot ou d'erreur de calibration dans la matrice de passage entre R_n et R_o , la trajectoire effectivement réalisée en pratique sera différente de celle attendue. La boucle fermée employée permet tout de même d'être robuste à ces erreurs de calibration et il est possible de quantifier cette robustesse par l'analyse de la condition de stabilité [3.62] en raisonnant dans l'espace articulaire.

3.3.2.1.3. Informations visuelles 2D 1/2

Comme on l'a déjà dit à maintes reprises, la prise en compte d'informations visuelles 3D repose sur l'hypothèse que ces données peuvent être mesurées de manière fiable. En pratique, elles sont plus sensibles aux erreurs de mesure que les informations visuelles 2D, puisqu'obtenues à partir de celles-ci et d'un calcul de pose n'ayant pas de propriété particulière de lissage. Il est donc intéressant de combiner des informations visuelles 2D et 3D pour gagner en robustesse aux erreurs de mesure tout en conservant de bonnes propriétés de découplage. Dans [MAL 98, MAL 99], la fonction de tâche est définie de la manière suivante :

$$\mathbf{e} = \left(x - x^* \quad y - y^* \quad \log(Z/Z^*) \quad \theta\mathbf{u} \right)^T \quad [3.67]$$

où :

- $(x \ y)^T$ et $(x^* \ y^*)^T$ sont respectivement les coordonnées courante et désirée d'un point caractéristique dans l'image ;
- Z/Z^* est le rapport entre les distances courante et désirée de ce point ;
- $\theta \mathbf{u}$ représente la rotation à réaliser.

Dans ce cas, on obtient comme loi de commande :

$$\mathcal{T}_{\mathbf{q}} = -\lambda \begin{pmatrix} Z\mathbf{L}_{e_v}^{-1} & -Z\mathbf{L}_{e_v}^{-1}\mathbf{L}_{e_{v\omega}} \\ \mathbf{0}_3 & \mathbb{I}_3 \end{pmatrix} \mathbf{e} \quad [3.68]$$

où (voir [3.16] et [3.45]) :

$$\mathbf{L}_{e_v} = \begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \\ 0 & 0 & -1 \end{pmatrix} \quad \text{et} \quad \mathbf{L}_{e_{v\omega}} = \begin{pmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \\ -y & x & 0 \end{pmatrix}$$

Un découplage satisfaisant est donc obtenu puisque la matrice de commande est triangulaire. De plus, la trajectoire du point caractéristique choisi sera rectiligne dans l'image. En sélectionnant correctement ce point (au centre de gravité de l'objet par exemple, ou le plus proche des limites de l'image, ce qui entraîne les mêmes inconvénients que ceux déjà décrits au début du paragraphe précédent), on peut conserver en général la présence de l'ensemble de l'objet dans l'image. Par ailleurs, à l'aide de résultats récents de reconstruction projective, il est possible d'utiliser ce schéma de commande sur des objets dont le modèle 3D est inconnu [MAL 00]. Le calcul de pose n'étant plus impliqué, on peut alors déterminer la forme analytique réelle de la matrice d'interaction (c'est-à-dire ne reposant pas sur les hypothèses fortes utilisées précédemment) et y faire intervenir les erreurs de calibration de la caméra. Grâce à la forme triangulaire de cette matrice, il est alors possible d'exhiber des conditions analytiques assurant les stabilités asymptotiques locale et globale du système [MAL 98, MAL 99].

Une variante à ce schéma est décrite dans [MOR 00]. La seule différence porte sur la troisième composante de \mathbf{e} qui prend explicitement en compte le fait que l'ensemble des points de l'objet doit rester, autant que possible, dans l'image. Cependant, la forme triangulaire de \mathbf{L}_{e_v} est alors perdue, ce qui rend difficile la détermination de conditions analytiques de stabilité en présence d'erreurs de calibration.

Un second schéma d'asservissement visuel 2D 1/2 est décrit dans [CHA 00] pour le cas d'une caméra embarquée. La fonction de tâche y est donnée par :

$$\mathbf{e} = \left({}^c \mathbf{P}_c^T \quad x - x^* \quad y - y^* \quad \theta u_z \right)^T \quad [3.69]$$

où $(x \ y)^T$ et $(x^* \ y^*)^T$ sont de nouveau les coordonnées courante et désirée d'un point caractéristique de l'image, et où u_z est la troisième composante de l'axe

\mathbf{u} de rotation entre R_{c^*} et R_c . A partir de [3.48], [3.16] et [3.41], on en déduit l'expression de la loi de commande associée :

$$\mathcal{T}_c = -\lambda \begin{pmatrix} {}^c\mathbf{R}_{c^*} & \mathbf{0}_3 \\ -\frac{1}{z}\mathbf{L}_{e_\omega}^{-1}\mathbf{L}_{e_{\omega v}} {}^c\mathbf{R}_{c^*} & \mathbf{L}_{e_\omega}^{-1} \end{pmatrix} \mathbf{e} \quad [3.70]$$

$$\text{où } \mathbf{L}_{e_{\omega v}} = \begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \\ 0 & 0 & 0 \end{pmatrix} \quad \text{et} \quad \mathbf{L}_{e_\omega} = \begin{pmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \\ l_1 & l_2 & l_3 \end{pmatrix}$$

(l_1 l_2 l_3) étant la troisième ligne de la matrice \mathbf{L}_ω donnée en [3.42].

Par rapport au cas précédent, la translation de la caméra suivra une trajectoire rectiligne, son orientation étant contrôlée de telle sorte que la trajectoire du point caractéristique suive une ligne droite dans l'image. Cette loi de commande est donc extrêmement intéressante en pratique. Si l'on sélectionne comme point caractéristique le point de l'objet le plus proche des limites de l'image, la discontinuité au changement de point de la loi de commande interviendra à présent sur les composantes de la vitesse de rotation. Par contre, on peut noter que la matrice de commande n'est plus que triangulaire par bloc, ce qui rend de nouveau difficile la détermination de conditions analytiques assurant la stabilité du système en présence d'erreurs de calibration. Il en est de même si l'on remplace dans la fonction de tâche θ_{u_z} par l'orientation d'un segment ou d'une droite dans l'image (voir paragraphe 3.2.4.2). On utilise alors davantage d'informations visuelles 2D et le seul changement intervenant dans la modélisation de la loi de commande consiste à remplacer les coefficients de la dernière ligne de $\mathbf{L}_{e_{\omega v}}$ et \mathbf{L}_{e_ω} par leurs nouvelles valeurs.

Enfin, dans [AND 99], les coordonnées Pluckeriennes de droites sont utilisées, ce qui aboutit également à un schéma d'asservissement visuel 2D 1/2. Les trajectoires obtenues dans l'espace et dans l'image ne semblent cependant pas aussi satisfaisantes que dans les deux cas décrits précédemment.

3.3.2.2. Cas où s est de dimension supérieure à 6 ($k > 6$)

Nous décrivons à présent les différents choix possibles pour \mathbf{C} et $\widehat{\mathbf{L}}_s$ (voir [3.58]) quand les informations visuelles choisies sont redondantes ($k > 6$). A l'exception des travaux décrits en [MAR 96b] où les coordonnées de plusieurs points 3D sont pris en compte, ce cas ne concerne que le choix d'informations visuelles 2D puisque la sélection de six informations visuelles indépendantes est alors problématique.

On rappelle que \mathbf{C} doit être une matrice constante, de dimension $6 \times k$, et de rang 6. Le choix le plus simple consiste à choisir pour \mathbf{C} la pseudo-inverse d'une

approximation de la valeur de la matrice d'interaction à la position désirée :

$$\mathbf{C} = \widehat{\mathbf{L}}_{\mathbf{s}=\mathbf{s}^*}^+ \quad [3.71]$$

Comme on l'a vu dans la première partie de ce chapitre, la matrice d'interaction dépend de la valeur des informations visuelles choisies et de la profondeur entre la caméra et les primitives correspondantes de la scène. Le calcul de \mathbf{C} nécessite donc de connaître la valeur de \mathbf{s}^* et des paramètres de profondeur à la position désirée. Si le modèle 3D de l'objet est disponible, ceux-ci peuvent se calculer aisément par un calcul de pose en utilisant l'image désirée. Autrement, ils sont en général déterminés lors de la spécification même de la tâche.

En utilisant [3.71], la loi de commande [3.58] s'écrit :

$$\mathcal{T}_{\mathbf{q}} = -\lambda \left(\widehat{\mathbf{L}}_{\mathbf{s}=\mathbf{s}^*}^+ \widehat{\mathbf{L}}_{\mathbf{s}} \right)^{-1} \mathbf{e} \quad [3.72]$$

et, en choisissant $\widehat{\mathbf{L}}_{\mathbf{s}=\mathbf{s}^*}$ pour approximer $\widehat{\mathbf{L}}_{\mathbf{s}}$, on obtient finalement :

$$\mathcal{T}_{\mathbf{q}} = -\lambda \mathbf{e} = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}=\mathbf{s}^*}^+ (\mathbf{s} - \mathbf{s}^*) \quad [3.73]$$

Il faut souligner que, même si \mathbf{e} est parfaitement régulée (i.e. $\mathbf{e} = \mathbf{0}$), cela n'implique pas forcément que la tâche visuelle soit réalisée (i.e. $\mathbf{s} = \mathbf{s}^*$). En effet, l'ensemble des configurations telles que :

$$(\mathbf{s} - \mathbf{s}^*) \in \text{Ker } \mathbf{C} \quad [3.74]$$

aboutit à annuler \mathbf{e} sans annuler $(\mathbf{s} - \mathbf{s}^*)$. Il faut donc veiller lors de la sélection des informations visuelles à ne pas créer de minima locaux potentiels dans l'espace de travail. A titre d'exemple, si l'on considère un carré centré parallèle au plan image, on peut montrer qu'en choisissant pour informations visuelles les coordonnées dans l'image des quatre coins de ce carré, les configurations correspondant à des minima locaux sont telles que la caméra observe le carré sur sa tranche. Les quatre points sont alors alignés dans l'image et l'on est alors en présence de cas dégénérés situés bien évidemment en dehors de l'espace de travail.

Par ailleurs, la condition de stabilité [3.60] s'écrit à présent :

$$\widehat{\mathbf{L}}_{\mathbf{s}=\mathbf{s}^*}^+ \mathbf{L}_{\mathbf{s}} > 0 \quad [3.75]$$

Même si $\widehat{\mathbf{L}}_{\mathbf{s}=\mathbf{s}^*}$ est parfaitement estimée, cette condition de positivité n'est assurée que dans un voisinage autour de la position désirée. De même, le comportement exponentiel découplé de \mathbf{e} ne sera assuré qu'à cette position désirée.

Il est donc possible, si la position initiale de la caméra est fortement éloignée de sa position désirée, que les trajectoires obtenues dans l'image soient peu satisfaisantes, voire ne permettent pas d'atteindre la convergence du système (voir figure 3.10.a). En pratique, ces cas de figure ne se rencontrent que si de forts mouvements en rotation sont nécessaires [CHA 98].

Pour tenter de remédier à ces problèmes, il est fréquent de choisir pour \mathbf{C} non plus une matrice constante, mais directement la pseudo-inverse d'une estimation de la valeur courante de la matrice d'interaction :

$$\mathbf{C} = \widehat{\mathbf{L}}_{\mathbf{s}}^+ \quad [3.76]$$

On obtient alors :

$$\mathcal{T}_{\mathbf{q}} = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}}^+ (\mathbf{s} - \mathbf{s}^*) \quad [3.77]$$

Il est à présent nécessaire d'effectuer à chaque itération de la loi de commande soit une estimation des paramètres 3D intervenant dans la matrice d'interaction, soit une estimation numérique en ligne des éléments de cette matrice (voir paragraphe 3.2.4.4). En absence de lissage temporel dans le calcul de cette matrice, le comportement du système sera donc moins stable que dans le cas précédent.

De plus, la condition de convergence [3.60] ne peut plus s'appliquer puisqu'il faudrait prendre en compte dans le calcul de $\dot{\mathbf{e}}$ les variations de \mathbf{C} (voir [3.55]), ce qui amènerait à des calculs quasiment inextricables. Si l'on considère le comportement de \mathbf{s} , on obtient :

$$\dot{\mathbf{s}} = -\lambda \mathbf{L}_{\mathbf{s}} \widehat{\mathbf{L}}_{\mathbf{s}}^+ (\mathbf{s} - \mathbf{s}^*) \quad [3.78]$$

Comme dans le cas précédent, il est donc impossible d'assurer la décroissance stricte de $\|\mathbf{s} - \mathbf{s}^*\|$ à chaque itération puisque la matrice $\mathbf{L}_{\mathbf{s}} \widehat{\mathbf{L}}_{\mathbf{s}}^+$, de dimension $k \times k$, est seulement de rang 6. Toutes les configurations telles que :

$$(\mathbf{s} - \mathbf{s}^*) \in \text{Ker } \widehat{\mathbf{L}}_{\mathbf{s}}^+ \quad [3.79]$$

correspondent en effet à des minima locaux attractifs, leur existence dans l'espace de travail étant démontrée dans [CHA 98] pour le cas très simple d'un carré.

L'inconvénient de cette méthode est qu'elle tente d'assurer $\dot{\mathbf{s}} = -\lambda(\mathbf{s} - \mathbf{s}^*)$ (qui fournit directement la loi de commande [3.77]), ce qui implique k contraintes alors que seuls 6 degrés de liberté sont disponibles. En d'autres termes, la "fonction de tâche" $(\mathbf{s} - \mathbf{s}^*)$ n'est pas ρ -admissible [SAM 91]. Par contre, l'avantage de cette méthode est qu'en cas de succès, elle fournit des trajectoires dans l'image très intéressantes. En effet, si \mathbf{s} est par exemple constitué de coordonnées de points de l'image, les trajectoires attendues de ces points seront des lignes

droites (voir figure 3.10.b). En pratique, les trajectoires effectivement réalisées ne seront pas forcément aussi sympathiques (puisque le comportement réel est donné par [3.78]). En outre, la trajectoire qu'effectuera le robot pour réaliser ces mouvements dans l'image n'est pas forcément souhaitable [CHA 98].

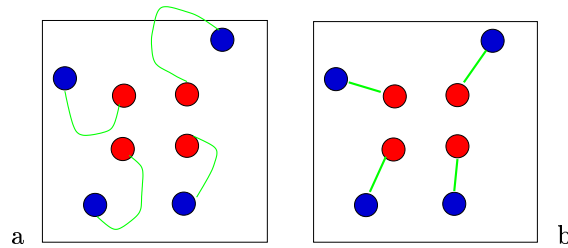


Figure 3.10. a) trajectoire possible dans l'image en choisissant $\mathbf{C} = \widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+$,
b) trajectoire attendue dans l'image en choisissant $\mathbf{C} = \widehat{\mathbf{L}}_{\mathbf{s}}^+$

Finalement, on trouve d'autres méthodes dans la littérature, basées par exemple sur l'utilisation de $\widehat{\mathbf{L}}_{\mathbf{s}}^T$ dans la loi de commande (au lieu de $\widehat{\mathbf{L}}_{\mathbf{s}|\mathbf{s}=\mathbf{s}^*}^+$ ou $\widehat{\mathbf{L}}_{\mathbf{s}}^+$) [HAS 93b]. Les avantages de ces méthodes par rapport à celles décrites précédemment ne semblent cependant pas évidents, puisqu'elles ne présentent pas de bonnes propriétés de découplage.

Pour conclure cette partie, on peut souligner l'importance du conditionnement de la matrice d'interaction et de la matrice de combinaison dans le comportement du système [FED 89, NEL 94b, SHA 95]. Un bon conditionnement de la première donne une bonne sensibilité au système, alors qu'un bon conditionnement de la seconde entraîne une bonne robustesse de la commande aux erreurs de mesure. Même si l'ensemble des méthodes présentées ci-dessus fournit des résultats pratiques très satisfaisants, de nombreux travaux restent à faire pour déterminer quelles sont les informations visuelles les plus pertinentes.

3.3.3. Tâches hybrides

Nous considérons à présent le cas où les k informations visuelles choisies ne contraignent pas l'ensemble des degrés de liberté du système. Les tâches visuelles associées aux k contraintes peuvent alors être classées en fonction de la liaison virtuelle, extension du concept des liaisons entre solides, entre le capteur et son environnement.

3.3.3.1. Liaisons virtuelles

Les contraintes $\mathbf{s}(\mathbf{r}(t)) - \mathbf{s}^* = 0$ induites par les informations visuelles définissent, lorsqu'elles sont réalisées, une **liaison virtuelle** entre le robot et son

environnement. Comme $\dot{\mathbf{s}} = 0$ est une conséquence immédiate de $\mathbf{s}(\mathbf{r}(t)) = \mathbf{s}^*$, l'ensemble \mathcal{S}^* des mouvements laissant \mathbf{s} invariant, soit :

$$\mathcal{S}^* = \text{Ker } \mathbf{L}_s \quad [3.80]$$

permet de caractériser complètement une liaison virtuelle.

A une situation \mathbf{r} où ces contraintes sont satisfaites, la dimension N de \mathcal{S}^* est appelée la *classe* de la liaison virtuelle au point \mathbf{r} . Soit $m = 6 - N$. Lorsque $m = k$, les k contraintes introduites par les informations visuelles sont indépendantes. Intuitivement, la dimension de \mathbf{s} correspond alors au nombre m de degrés de liberté que l'on peut et souhaite contrôler à partir de \mathbf{s} . Comme on l'a déjà vu précédemment, le cas où les informations visuelles sont redondantes ($k > m$) peut également se rencontrer.

Sur la figure 3.11 sont répertoriées les liaisons mécaniques sans frottement les plus classiques. La classe de chaque liaison et le nombre de degrés de liberté non contraints (en translation (T) ou en rotation (R)) qui permettent cette classification y sont indiqués. Avec un capteur de vision, il est possible de réaliser l'ensemble de ces liaisons. Par exemple, le cas de la liaison rigide (contraignant les 6 degrés de liberté du robot) a déjà été étudié en détail dans le paragraphe précédent. Pour les autres liaisons, des exemples de motifs, construits à partir des primitives géométriques les plus usuelles (points, droites, cylindres, etc.) sont donnés dans [CHA 90, CHA 93a].

A titre d'exemple, nous présentons le cas de la liaison rotule. Celle-ci peut être réalisée si la caméra observe une sphère dont le centre M est situé sur son axe optique. L'image de la sphère est alors un cercle centré, et si l'on sélectionne dans \mathbf{s} l'aire a du cercle et les coordonnées $(x_g \ y_g)^T$ de son centre de gravité ($\mathbf{s} = (a \ x_g \ y_g)^T$), la matrice d'interaction a pour valeur (voir [3.34]) :

$$\mathbf{L}_s = \begin{pmatrix} 0 & 0 & 2a/Z_g & 0 & 0 & 0 \\ -1/Z_g & 0 & 0 & 0 & -(1+r^2) & 0 \\ 0 & -1/Z_g & 0 & 1+r^2 & 0 & 0 \end{pmatrix} \quad [3.81]$$

où $1/Z_g = Z_0/(Z_0^2 - R^2)$ et $r^2 = R^2/(Z_0^2 - R^2)$, r étant le rayon du cercle dans l'image, R le rayon de la sphère, et Z_0 la distance de la caméra à la sphère. En exprimant la matrice d'interaction au centre de la sphère à l'aide de la formule de changement de repère des torseurs donnée en [3.6], on obtient :

$$\begin{aligned} \mathbf{L}_s &= \begin{pmatrix} 0 & 0 & 2a/Z_g & 0 & 0 & 0 \\ -1/Z_g & 0 & 0 & 0 & -(1+r^2) & 0 \\ 0 & -1/Z_g & 0 & 1+r^2 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbb{I}_3 & \tilde{\mathbf{M}} \\ \mathbf{0}_3 & \mathbb{I}_3 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 2a/Z_g & 0 & 0 & 0 \\ -1/Z_g & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/Z_g & 0 & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

Nom de la liaison	Classe	T	R	Symbole géométrique
Rigide	0	0	0	
Prismatique	1	1	0	
Rotoïde	1	0	1	
Pivot glissant	2	1	1	
Appui plan	3	2	1	
Rotule	3	0	3	
Linéaire rectiligne	4	2	2	
Linéaire annulaire	4	1	3	
Ponctuelle	5	2	3	

Figure 3.11. Liaisons mécaniques

qui nous donne bien la forme voulue pour \mathcal{S}^* :

$$\mathcal{S}^* = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad [3.82]$$

On peut bien sûr sélectionner des informations visuelles 3D pour réaliser cette liaison (si des mesures 3D sont disponibles). En l'occurrence, il suffit de choisir dans \mathbf{s} les trois coordonnées du centre de la sphère.

Nous détaillons à présent comment définir une fonction de tâche hybride combinant une tâche visuelle contrôlant $m (< 6)$ degrés de liberté et une tâche secondaire.

3.3.3.2. Fonction de tâche hybride

Très souvent, la régulation d'une tâche visuelle n'est pas l'unique objectif visé et cette tâche doit être combinée avec une autre, telle par exemple un suivi d'une trajectoire ou l'évitement des butées articulaires du robot. En effet, excepté le cas où la tâche visuelle consiste à réaliser une liaison rigide, les autres tâches réalisent des liaisons virtuelles de classe non nulle, c'est-à-dire que des degrés de liberté ne sont pas contraints par la liaison. Par exemple, un mouvement de translation dans l'axe d'une liaison prismatique conserve cette liaison, et il peut être souhaitable d'utiliser ce degré de liberté disponible pour réaliser un second objectif.

La combinaison des deux objectifs est parfois effectuée par une simple sommation [NEL 95]. La loi de commande réalise alors un compromis qui peut aboutir à ce qu'aucun des objectifs initiaux ne soit atteint. On rappelle ici brièvement une autre approche plus élégante, détaillée notamment dans [SAM 91]. Elle consiste à considérer la tâche visuelle comme prioritaire et à exprimer le second objectif comme une fonction de coût à minimiser sous la contrainte que la tâche visuelle soit réalisée. L'utilisation de cette approche commence à être courante en asservissement visuel. On trouve dans [CHA 90, COS 95, BER 99] le cas où la tâche secondaire consiste à effectuer des suivis de trajectoires. Ceux-ci peuvent être utiles dans des applications d'inspection et de contrôle de conformité, mais aussi pour effectuer la reconstruction 3D des objets considérés [BOU 93, CHA 96] ou assurer un positionnement correct du système [COL 99a]. L'évitement des butées et des singularités est traité dans [MAR 96a]. Les tâches secondaires peuvent également être des tâches visuelles, pour tenter d'éviter des occultations par exemple [MAR 98], ou

construites à partir de mesures fournies par d'autres capteurs extéroceptifs. Par exemple, la combinaison d'une tâche de positionnement par asservissement visuel et d'une tâche d'évitement d'obstacles à l'aide d'un capteur proximétrique de type laser a été étudiée dans [CAD 99] dans le cadre de la robotique mobile.

On note \mathbf{e}_1 la fonction de tâche visuelle et h_s la fonction de coût à minimiser, de gradient $\mathbf{e}_2^T = \frac{\partial h_s}{\partial \mathbf{r}}$. La fonction \mathbf{e}_1 est de dimension $m = n - N \leq k$ où n est le nombre de degrés de liberté du système, N la classe de la liaison virtuelle souhaitée, et k le nombre d'informations visuelles utilisées dans \mathbf{s} . Cette fonction de tâche s'écrit toujours :

$$\mathbf{e}_1 = \mathbf{C} (\mathbf{s} - \mathbf{s}^*) \quad [3.83]$$

où \mathbf{C} est à présent de dimension $m \times k$ et de rang plein m afin que les m composantes de \mathbf{e}_1 soient indépendantes et contrôlent les m degrés de liberté souhaités. Si \mathbf{C} est choisie constante, le jacobien \mathbf{J}_1 de \mathbf{e}_1 , de dimension $m \times n$ et de rang plein m , est donné par :

$$\mathbf{J}_1 = \frac{\partial \mathbf{e}_1}{\partial \mathbf{r}} = \mathbf{C} \mathbf{L}_s \quad [3.84]$$

et on peut noter que $\text{Ker } \mathbf{J}_1 = \text{Ker } \mathbf{L}_s$.

Une fonction de tâche \mathbf{e} réalisant la minimisation de h_s sous la contrainte $\mathbf{e}_1 = 0$ s'écrit sous la forme [SAM 91] :

$$\mathbf{e} = \mathbf{W}^+ \mathbf{e}_1 + (\mathbb{I}_n - \mathbf{W}^+ \mathbf{W}) \mathbf{e}_2 \quad [3.85]$$

où \mathbf{W} est une matrice de dimension $m \times n$, de rang plein m et telle que $\text{Ker } \mathbf{W} = \text{Ker } \mathbf{L}_s$. La matrice $(\mathbb{I}_n - \mathbf{W}^+ \mathbf{W})$ est donc un opérateur de projection orthogonale sur le noyau de \mathbf{L}_s , ce qui signifie que :

$$(\mathbb{I}_n - \mathbf{W}^+ \mathbf{W}) \mathbf{x} \in \text{Ker } \mathbf{L}_s, \quad \forall \mathbf{x} \in \mathbb{R}^n \quad [3.86]$$

Quelle que soit la fonction à minimiser, et donc la tâche secondaire \mathbf{e}_2 , seuls des mouvements ne perturbant pas \mathbf{e}_1 seront donc appliqués, ce qui implique aussi que h_s n'atteindra pas forcément sa valeur minimale.

Cependant, comme la valeur exacte de la matrice d'interaction est en général inconnue, $\widehat{\mathbf{W}}$ ne pourra être construit qu'à partir d'une approximation ou d'une estimation $\widehat{\mathbf{L}}_s$. Si le noyau de \mathbf{W} est différent de celui de \mathbf{L}_s , la tâche secondaire peut alors entraîner des perturbations sur la réalisation de \mathbf{e}_1 . En pratique, ces perturbations ne s'avèrent guère pénalisantes, à moins que l'estimation de la matrice d'interaction ne soit complètement erronée.

Pour construire la matrice \mathbf{W} , le cas le plus simple se produit quand \mathbf{L}_s est de rang plein $m = k$. On peut alors en effet prendre directement $\mathbf{W} = \widehat{\mathbf{L}}_s$. Autrement, les m lignes de \mathbf{W} peuvent être formées par les m vecteurs de base

du sous-espace engendré par $\widehat{\mathbf{L}}_s$. On peut noter que si \mathbf{L}_s est de rang n (c'est-à-dire que la tâche visuelle contraint les n degrés de liberté du système), on peut choisir $\mathbf{W} = \mathbb{I}_n$. On ne peut alors évidemment pas prendre en compte de tâche secondaire, puisqu'on a alors $\mathbf{e} = \mathbf{e}_1$. Cela indique aussi que la forme de la fonction de tâche présentée en [3.85] est une généralisation de celle donnée précédemment en [3.53].

En effectuant la même analyse que celle décrite au début du paragraphe 3.3.2, on obtient comme loi de commande :

$$\mathcal{T}_q = \left(\frac{\partial \mathbf{e}}{\partial \mathbf{r}} \right)^{-1} \left(-\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t} \right) \quad [3.87]$$

et comme condition suffisante de stabilité :

$$\frac{\partial \mathbf{e}}{\partial \mathbf{r}} \left(\frac{\partial \mathbf{e}}{\partial \mathbf{r}} \right)^{-1} > 0 \quad [3.88]$$

Vu la difficulté à calculer $\frac{\partial \mathbf{e}}{\partial \mathbf{r}}$ en pratique, on peut montrer [SAM 91] que la condition [3.88] est en général respectée si l'on a :

$$\frac{\partial \mathbf{e}_1}{\partial \mathbf{r}} \mathbf{W}^+ > 0 \quad [3.89]$$

On peut alors fixer $\frac{\partial \mathbf{e}}{\partial \mathbf{r}} = \mathbb{I}_n$, ce qui donne la loi de commande :

$$\mathcal{T}_q = -\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t} \quad [3.90]$$

Comme pour le cas de la liaison rigide vue en 3.3.2, on a tout intérêt à choisir $\mathbf{C} = \mathbb{I}_m$ si les informations visuelles ne sont pas redondantes ($k = m$). On peut d'ailleurs signaler qu'il est beaucoup moins délicat de sélectionner des informations visuelles 2D non redondantes pour les autres liaisons que la liaison rigide (cf. à titre d'exemple le cas de la liaison rotule décrit au paragraphe précédent). En choisissant $\mathbf{C} = \mathbb{I}_m$, la condition [3.89] s'écrit simplement $\mathbf{L}_s \mathbf{W}^+ > 0$, qui est respectée si l'estimation utilisée pour construire \mathbf{W} est correcte. On retrouvera de plus une décroissance exponentielle pour chaque composante de \mathbf{s} . Dans le cas où les informations visuelles sont redondantes ($k > m$), on peut choisir :

$$\mathbf{C} = \mathbf{W} \widehat{\mathbf{L}}_{s|s=s^*}^+ \quad [3.91]$$

Si \mathbf{W} est également construite à partir de $\widehat{\mathbf{L}}_{s|s=s^*}$ (et donc constante), la condition [3.89] s'écrit :

$$\mathbf{W} \widehat{\mathbf{L}}_{s|s=s^*}^+ \mathbf{L}_s \mathbf{W}^+ > 0 \quad [3.92]$$

qui ne peut au mieux être respectée qu'au voisinage des configurations où $\mathbf{s} = \mathbf{s}^*$. En pratique, on peut également considérer des matrices \mathbf{W} et \mathbf{C} calculées à chaque itération à partir d'une estimation de la valeur courante de la matrice d'interaction, mais ce choix ne permet plus de déterminer aisément la forme analytique de $\frac{\partial \mathbf{e}_1}{\partial \mathbf{r}}$.

Finalement, si la tâche secondaire permet de connaître $\frac{\partial \mathbf{e}_2}{\partial t}$, on peut choisir :

$$\frac{\widehat{\partial \mathbf{e}}}{\partial t} = \mathbf{W}^+ \frac{\widehat{\partial \mathbf{e}_1}}{\partial t} + (\mathbb{I}_n - \mathbf{W}^+ \mathbf{W}) \frac{\partial \mathbf{e}_2}{\partial t} \quad [3.93]$$

où le terme $\frac{\widehat{\partial \mathbf{e}_1}}{\partial t}$, s'il est correctement estimé, permet de réduire les éventuelles erreurs de traînage. Nous décrivons à présent comment estimer ce terme.

3.3.4. Poursuite de cibles

On considère dans ce court paragraphe le cas d'une caméra embarquée observant un objet mobile. Les principes décrits ci-dessous sont cependant également valides pour une caméra déportée animée d'un mouvement inconnu.

Une bonne partie des travaux en asservissement visuel traitant de la poursuite de cible considère le mouvement de l'objet comme une perturbation à rejeter le plus rapidement et le plus efficacement possible [PAP 93, GAN 98, GAN 99]. D'autres travaux utilisent des connaissances a priori sur la trajectoire de l'objet [ALL 93, HAS 95, RIZ 96]. Par ailleurs, l'utilisation d'un intégrateur est classique en automatique pour supprimer les erreurs de traînage. Notons \mathbf{I}_k l'estimation de $\frac{\partial \mathbf{e}_1}{\partial t}$ à l'itération k . On a alors :

$$\begin{aligned} \mathbf{I}_{k+1} &= \mathbf{I}_k + \mu \mathbf{e}_{1k} \text{ avec } \mathbf{I}_0 = 0 \\ &= \mu \sum_{j=0}^k \mathbf{e}_{1j} \end{aligned} \quad [3.94]$$

où μ est le gain de l'intégrateur. Cette technique ne permet de traiter correctement que les cas où l'objet est à vitesse constante. On a en effet $\mathbf{I}_{k+1} = \mathbf{I}_k$ si et seulement si $\mathbf{e}_{1k} = 0$. Cela implique que les erreurs de traînage ne seront pas entièrement supprimées si le mouvement de l'objet est plus complexe.

D'autres classes de méthodes consistent à estimer le plus fidèlement possible la vitesse de l'objet dans l'image. En effet, s'il est possible de mesurer la vitesse de la caméra, à l'aide par exemple de l'odométrie du robot porteur de cette caméra, une estimation de la vitesse de l'objet dans l'image est donnée par :

$$\frac{\widehat{\partial \mathbf{e}_1}}{\partial t} = \widehat{\mathbf{e}}_1 - \left(\frac{\widehat{\partial \mathbf{e}_1}}{\partial \mathbf{r}} \right) \mathcal{T}_c \quad [3.95]$$

où $\frac{\partial \widehat{\mathbf{e}}_1}{\partial \mathbf{r}} = \mathbf{C}\widehat{\mathbf{L}}_s$ et où $\widehat{\mathbf{e}}_1$ est par exemple mesuré à l'itération k par $\widehat{\mathbf{e}}_{1k} = (\mathbf{e}_{1k} - \mathbf{e}_{1k-1})/\Delta t$, Δt étant la période d'échantillonnage du système. On aboutit ainsi à un système de commande adaptatif indirect et on peut ensuite utiliser un filtre de Kalman (par exemple) pour lisser cette estimation. Dans [COR 93], un tel filtre basé sur un simple modèle d'état à vitesse constante est présenté. Dans [CHA 93b], un modèle d'état à accélération constante et bruit corrélé a été choisi. Enfin, l'algorithme dit du GLR (*Generalized Likelihood Ratio*) est utilisé dans [BEN 95, BEN 96] afin de détecter, d'estimer et de compenser d'éventuels sauts en direction ou en amplitude de la vitesse de l'objet.

3.4. Autres capteurs extéroceptifs

L'ensemble des principes décrits dans ce chapitre est valide quel que soit le capteur extéroceptif considéré. La seule particularité porte sur la modélisation de la matrice d'interaction entre le capteur considéré et son environnement [SAM 91].

Considérons par exemple un capteur proximétrique à champ étroit qui fournit la distance Z entre ce capteur et l'objet le plus proche dans la direction du capteur (voir figure 3.12). Si l'on suppose la surface de l'objet perpendiculaire à l'axe du capteur, la matrice d'interaction associée à Z a la forme :

$$\mathbf{L}_Z = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \quad [3.96]$$

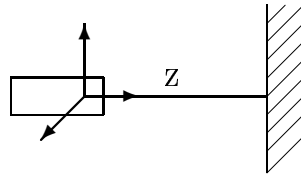


Figure 3.12. Modélisation d'un capteur proximétrique

On peut alors réaliser diverses tâches robotiques comme l'évitement d'obstacles ou la plupart des liaisons virtuelles que l'on a vu précédemment en sélectionnant le nombre et la direction adéquats de capteurs. Par exemple, si l'on souhaite réaliser une liaison appui plan, on peut choisir la configuration présentée sur la figure 3.13. Exprimée dans le repère $R_O = (O, \vec{x}, \vec{y}, \vec{z})$, la matrice d'interaction \mathbf{L}_{Z_i} associée à chaque capteur S_i est donnée par (voir [3.6]) :

$$\mathbf{L}_{Z_i} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbb{I}_3 & -\tilde{\mathbf{S}}_i \\ 0 & \mathbb{I}_3 \end{pmatrix} \quad [3.97]$$

soit finalement :

$$\mathbf{L}_{Z_i} = \begin{pmatrix} 0 & 0 & -1 & -Y_i & X_i & 0 \end{pmatrix} \quad [3.98]$$

où $(X_i \ Y_i \ 0)^T$ sont les coordonnées de S_i dans R_O . On vérifie ainsi que la matrice d'interaction intégrant les quatre capteurs est de rang 3 et de noyau :

$$\mathcal{S}^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad [3.99]$$

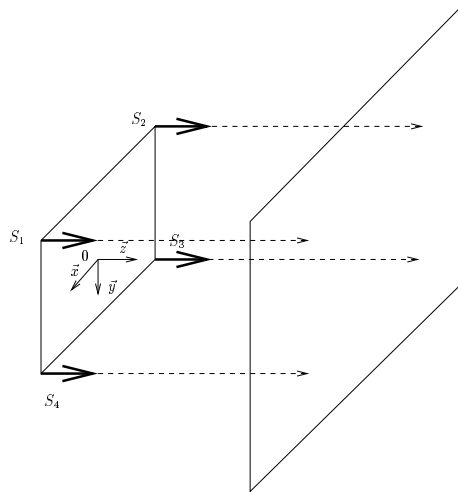


Figure 3.13. *Liaison appui plan*

L'utilisation de capteurs d'effort a également fait l'objet de nombreux travaux [SAM 91, BOI 88]. On peut finalement citer [MOT 92, KHA 96a] où le capteur considéré est constitué d'une caméra liée rigidement à des plans-laser.

3.5. Conclusion

Dans ce chapitre, nous n'avons considéré que le cas le plus général où le système robotique à contrôler a six degrés de liberté. La sélection d'informations visuelles pertinentes pour réaliser la tâche est alors le problème le plus difficile, surtout si l'on s'intéresse à la liaison rigide. De nombreuses simplifications interviennent si le système dispose de moins de degrés de liberté. A titre d'exemple, le contrôle de l'orientation d'une caméra montée sur une tour "pan-tilt" pour une tâche de centrage et de poursuite d'un objet mobile ne présente aucune difficulté en ce qui concerne la modélisation de la tâche. Aucune information 3D n'est même nécessaire. La difficulté principale porte

alors sur les aspects de traitement d'images afin d'appréhender des objets réels complexes.

En effet, pendant de nombreuses années, les techniques d'asservissement visuel n'ont pu être appliquées qu'à des objets extrêmement simples (fournissant en général des images binaires), en raison de la cadence trop faible de traitement sur des images complexes. Les progrès algorithmiques, notamment dans le domaine de l'analyse du mouvement, mais surtout les gains énormes de puissance de calcul, permettent à présent de considérer des applications réelles tournant à une cadence proche de la cadence vidéo [CRE 98a, CRE 98b, TON 97, MAR 99, VIN 00]. On peut également signaler les travaux récents où les informations visuelles ne sont plus de type géométriques, mais décrivent une information de mouvement dans la séquence d'images [GRO 95, CRE 97, CRE 98b].

Les perspectives principales en asservissement visuel portent sur la prise en compte d'objets inconnus, ne nécessitant pas de connaissance 3D a priori. Les travaux dans le domaine de la géométrie projective ont déjà donné lieu à des résultats prometteurs [MAL 98, MAL 00, RUF 99]. En pratique, si l'apprentissage de l'image désirée est impossible, l'obtention de la consigne lorsque l'on dispose d'un système grossièrement calibré est notamment très délicate. Par ailleurs, les aspects de planification de trajectoires dans l'espace du capteur [HOS 95b] font actuellement l'objet de fructueux travaux [MEZ 00]. Dans le même ordre d'idée, les aspects d'enchaînement de tâches visuelles mériteraient de plus amples développements. On pourrait alors élargir l'aspect très local des techniques actuelles. De même, la fusion de mesures fournies par plusieurs capteurs extéroceptifs, de nature différente ou non, au sein de lois de commande devraient permettre de traiter de nombreuses nouvelles applications [CAD 99, MAL 01].

Enfin, l'utilisation de l'asservissement visuel en robotique mobile connaît depuis quelques années un essor important, notamment pour les applications de conduite automatique de véhicules [DIC 91, PIS 93, DEB 96, KHA 96b, TSA 98, SWA 99]. En sus des problèmes de traitement d'image, la difficulté essentielle réside alors dans l'élaboration de lois de commande prenant en compte les contraintes non holonomes de ce type de robots.

3.6. Remerciements

L'auteur tient à remercier Christophe Collewet, Armel Crétual, Etienne Dombre, Bernard Espiau, Grégory Flandin, Radu Horaud, Wissama Khalil, Youcef Mezouar, Ezio Malis, Eric Marchand, Patrick Rives et Fabien Spindler, premiers lecteurs de ce chapitre, pour leurs remarques constructives ou leur absence de remarque destructive.

3.7. Bibliographie

- [ALL 93] ALLEN P.K., YOSHIMI B., TIMCENKO A., MICHELMAN P., «Automated tracking and grasping of a moving object with a robotic hand-eye system », *IEEE Trans. on Robotics and Automation*, vol. 9, n° 2, p. 152-165, 1993.
- [AND 99] ANDREFF N., Asservissement visuel à partir de droites et auto-étalonnage pince-caméra, Thèse de l'Institut National Polytechnique de Grenoble, INRIA Rhône-Alpes, 1999.
- [BEN 95] BENSALAH F., CHAUMETTE F., «Compensation of abrupt motion changes in target tracking by visual servoing», *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, Pittsburgh, p. 181-187, 1995.
- [BEN 96] BENSALAH F., Estimation du mouvement par vision active, Thèse de l'Université de Rennes 1, IRISA, 1996.
- [BER 99] BERRY F., Contournement d'objets par asservissement visuel, Thèse de l'Université Blaise Pascal de Clermont-Ferrand, LASMEA, 1999.
- [BEY 92] BEYER H., «Accurate calibration of CCD cameras », *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, Urbana Champaign, p. 96-101, 1992.
- [BOI 88] BOISSONNAT J.D., FAVERJON B., MERLET J.P., *Techniques de la robotique : perception et planification*, Hermès, Paris, 1988.
- [BOU 93] BOUKIR S., Reconstruction 3D d'un environnement statique à l'aide d'une caméra mobile commandable, Thèse de l'Université de Rennes 1, IRISA, 1993.
- [CAD 99] CADENAT V., Commande référencée multi-capteurs pour la navigation d'un robot mobile, Thèse de l'Université Paul-Sabatier de Toulouse, LAAS, 1999.
- [CHA 90] CHAUMETTE F., La relation vision-commande : théorie et applications à des tâches robotiques, Thèse de l'Université de Rennes 1, IRISA, 1990.
- [CHA 93a] CHAUMETTE F., RIVES P., ESPIAU B., «Classification and realization of the different vision-based tasks », In [HAS 93a], p. 199-228, 1993.
- [CHA 93b] CHAUMETTE F., SANTOS A., «Tracking a moving object by visual servoing », *Proc. 12th World Congress IFAC*, Sydney, vol. 9, p. 643-648, 1993.
- [CHA 96] CHAUMETTE F., BOUKIR S., BOUTHEMY P., JUVIN D., «Structure from controlled motion », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, n° 5, p. 492-504, 1996.
- [CHA 98] CHAUMETTE F., «Potential problems of stability and convergence in image-based and position-based visual servoing », In *The Confluence of Vision and Control, LNCIS*, vol. 237, p. 66-78, Springer Verlag, 1998.
- [CHA 00] CHAUMETTE F., MALIS E., «2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings », *IEEE Int. Conf. on Robotics and Automation, ICRA'00*, vol. 1, p. 630-635, San Francisco, 2000.
- [CIP 97] CIPOLLA R., HOLLINGHURST N., Visually guided grasping in unstructured environment. *Robotics and Autonomous Systems*, vol. 19, p. 337-346, 1997.
- [COL 99a] COLLEWET C., Contributions à l'élargissement du champ applicatif des asservissements visuels 2D, Thèse de l'Université de Rennes 1, CEMAGREF, 1999.
- [COL 99b] COLOMBO C., ALLOTTA B., «Image-based robot task planning and control using a compact visual representation », *IEEE Trans. on Systems, Man and Cybernetics*, vol. A-29, n° 1, p. 92-100, 1999.

- [COR 93] CORKE P.I., GOODS M.C., «Controller design for high performance visual servoing », *Proc. 12th World Congress IFAC*, Sydney, vol. 9, p. 395-398, 1993.
- [COS 95] COSTE-MANIÈRE E., COUVIGNOU P., KHOSLA P.K., «Visual servoing in the task-function framework: a contour following task », *Journal of Intelligent and Robotic Systems*, vol. 12, p. 1-21, 1995.
- [CRE 97] CRÉTUAL A., CHAUMETTE F., «Positioning a camera parallel to a plane using dynamic visual servoing », *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'97*, Grenoble, vol. 1, p. 43-48, 1997.
- [CRE 98a] CRÉTUAL A., CHAUMETTE F., «Image-based visual servoing by integration of dynamic measurements », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'98*, Louvain, vol. 3, p. 1994-2001, 1998.
- [CRE 98b] CRÉTUAL A., Asservissement visuel à partir d'informations de mouvement dans l'image, Thèse de l'Université de Rennes 1, IRISA, 1998.
- [DEB 96] DEBAIN C., Lois de commande pour le contrôle et la mobilité des machines agricoles, Thèse de l'Université Blaise Pascal de Clermont-Ferrand, LASMEA, 1996.
- [DEG 97] DEGUCHI K., «Direct interpretation of dynamic images and camera motion for visual servoing without image feature correspondence », *Journal of Robotics and Mechatronics*, vol. 9, n° 2, p. 104-110, 1997.
- [DEM 95] DEMENTHON D., DAVIS L.S., «Model-based object pose in 25 lines of code », *Int. Journal of Computer Vision*, vol. 15, n° 1/2, p. 123-141, 1995.
- [DHO 89] DHOME M., RICHTIN M., LAPRESTE J.T., RIVES G., «Determination of the attitude of 3D objects from a single perspective view », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, n° 12, p. 1265-1278, 1989.
- [DHO 90] DHOME M., LAPRESTE J.T., RIVES G., RICHTIN M., «Spatial localization of modelled objects of revolution in monocular perspective vision », *Proc. European Conf. on Computer Vision, ECCV'90*, Antibes, LNCS, vol. 427, p. 475-485, 1990.
- [DIC 91] DICKMANN E., CHRITIANS T., «Relative 3D state estimation for autonomous visual guidance of road vehicles », *Robotics and Autonomous Systems*, vol. 7, n° 2/3, p. 85-98, 1991.
- [DOR 95] DORNAIKA F., Contributions à l'intégration vision/robotique : calibrage, localisation, asservissement, Thèse de l'Institut National Polytechnique de Grenoble, INRIA Rhône-Alpes, 1995.
- [DRU 99] DRUMMOND T., CIPOLLA R., «Visual tracking and control using Lie algebra », *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'99*, Fort Collins, vol. 2, p. 652-657, 1999.
- [ESP 92] ESPIAU B., CHAUMETTE F., RIVES P., «A new approach to visual servoing in robotics », *IEEE Trans. on Robotics and Automation*, vol. 8, n° 6, p. 313-326, 1992.
- [ESP 93] ESPIAU B., «Effect of camera calibration errors on visual servoing in robotics », *Proc. Int. Symposium on Experimental Robotics, ISER'93*, Kyoto, 1993.
- [ESP 95] ESPIAU B., Asservissement visuel et commande de la distance focale. Rapport de recherche INRIA, n° 2613, 1995.
- [FAU 93] FAUGERAS O., *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, Cambridge, 1993.

- [FED 89] FEDDEMA J., LEE C., MITCHELL O., «Automatic selection of image features for visual servoing of a robot manipulator », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'89*, Scottsdale, vol. 2, p. 832-837, 1989.
- [GAN 98] GANGLOFF J., DE MATHELIN M., ABBA G., «6 DOF high speed dynamic visual servoing using GPC controllers », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'98*, Louvain, p. 1008-1013, 1998.
- [GAN 99] GANGLOFF J., Asservissements visuels rapides d'un robot manipulateur à six degrés de liberté, Thèse de l'Université Louis Pasteur de Strasbourg, LSIIT, 1999.
- [GRO 95] GROSSO E., TISTARELLI M., «Active/dynamic stereo vision », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, n° 11, p. 1117-1127, 1995.
- [HAG 95] HAGER G., CHANG W., MORSE A.S., «Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination », *IEEE Control Systems Magazine*, vol. 15, n° 1, p. 30-39, 1995.
- [HAR 89] HARALICK R., JOO H., LEE C.N., ZHUANG X., VAIDYA V., KIM M., «Pose estimation from corresponding point data », *IEEE Trans. on Systems, Man and Cybernetics*, vol. 19, n° 6, p. 1426-1445, 1989.
- [HAS 93a] HASHIMOTO K., ED., *Visual Servoing : Real Time Control of Robot manipulators based on visual sensory feedback*, Vol. 7 of *World Scientific Series in Robotics and Automated Systems*, World Scientific Press, Singapour, 1993.
- [HAS 93b] HASHIMOTO K., KIMURA H., «LQ Optimal and non linear approaches to visual servoing », In [HAS 93a], p. 165-198, 1993.
- [HAS 95] HASHIMOTO K., KIMURA H., «Visual servoing with non linear observer », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'95*, Nagoya, p. 484-489, 1995.
- [HAS 96] HASHIMOTO K., EBINE T., KIMURA H., «Visual servoing with hand-eye manipulator-Optimal control approach », *IEEE Trans. on Robotics and Automation*, vol. 12, n° 5, p. 766-774, 1996.
- [HOR 89] HORAUD R., CONIO B., LEBoulLEUX O., LACOLLE B., «An analytic solution for the perspective 4-points problem », *Computer Vision, Graphics, and Image Processing*, vol. 47, n° 1, p. 33-44, 1989.
- [HOR 95] HORAUD R., MONGA O., *Vision par ordinateur, outils fondamentaux*, Hermès, Paris, 2ème édition, 1995.
- [HOR 98] HORAUD R., DORNAIKA F., ESPIAU B., «Visually guided object grasping », *IEEE Trans. on Robotics and Automation*, vol. 14, n° 4, p. 525-532, 1998.
- [HOS 94] HOSODA K., ASADA M., «Versatile visual servoing without knowledge of true jacobian », *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'94*, Munich, 1994.
- [HOS 95a] HOSODA K., MORIYAMA H., ASADA M., «Visual servoing utilizing zoom mechanism », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'95*, Nagoya, 1995.
- [HOS 95b] HOSODA K., SAKAMOTO K., ASADA M., «Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3D reconstruction », *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, Pittsburgh vol. 3, p.29-34, 1995.

- [HUT 96] HUTCHINSON S., HAGER G., CORKE P.I., «A tutorial on visual servo control », *IEEE Trans. on Robotics and Automation*, vol. 12, n° 5, p. 651-670, 1996.
- [JAG 97] JÄGERSAND M., FUENTES O., NELSON R., «Experimental evaluation of uncalibrated visual servoing for precision manipulation », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'97*, Albuquerque, vol. 3, p. 2874-2880, 1997.
- [KEL 96] KELLY R., «Robust asymptotically stable visual servoing of planar robots », *IEEE Trans. on Robotics and Automation*, vol. 12, n° 5, p. 759-766, 1996.
- [KHA 96a] KHADRAOUI D., MOTYL G., MARTINET P., GALLICE J., CHAUMETTE F., «Visual servoing in robotics scheme using a camera/laser-stripe sensor », *IEEE Trans. on Robotics and Automation*, vol. 12, n° 5, p. 743-750, 1996.
- [KHA 96b] KHADRAOUI D., La commande référencée vision pour le guidage automatique de véhicules, Thèse de l'Université Blaise Pascal de Clermont-Ferrand, LASMEA, 1996.
- [KHA 99] KHALIL W., DOMBRE E., *Modélisation, identification, et commande des robots*, Hermès, Paris, 2ème édition, 1999.
- [KIN 94] KINOSHITA K., DEGUCHI K., «Simultaneous determination of camera pose and intrinsic parameters by visual servoing », *Proc. Int. Conf. on Pattern Recognition, ICPR'94*, Jerusalem, vol.1, p. 285-290, 1994.
- [LOW 87] LOWE D.G., «Three-dimensional object recognition from single two-dimensional images », *Artificial Intelligence*, vol. 31, p. 355-394, 1987.
- [MA 93] DE MA S., «Conics-based stereo, motion estimation, and pose determination », *Int. Journal of Computer Vision*, vol. 10, n° 1, p. 7-25, 1993.
- [MAL 98] MALIS E., Contributions à la modélisation et à la commande en asservissement visuel, Thèse de l'Université de Rennes 1, IRISA, 1998.
- [MAL 99] MALIS E., CHAUMETTE F., BOUDET S., «2 1/2 D visual servoing », *IEEE Trans. on Robotics and Automation*, vol. 15, n° 2, p. 238-250, 1999.
- [MAL 00] MALIS E., CHAUMETTE F., «2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement », *Int. Journal of Computer Vision*, vol. 37, n° 1, p. 79-97, 2000.
- [MAL 01] MALIS E., MOREL G., CHAUMETTE F., «Robot control from disparate multiple sensors », *Int. Journal of Robotics Research*, vol. 20, n° 5, p. 364-378, 2001.
- [MAR 96a] MARCHAND E., CHAUMETTE F., RIZZO A., «Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing », *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'96*, Osaka, vol. 3, p. 1083-1090, 1996.
- [MAR 98] MARCHAND E., HAGER G., «Dynamic sensor planning in visual servoing », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'98*, Louvain, vol. 3, p. 1988-1993, 1998.
- [MAR 99] MARCHAND E., BOUTHEMY P., CHAUMETTE F., MOREAU V., «Robust real-time visual tracking using a 2D-3D model-based approach », *Proc. IEEE Int. Conf. on Computer Vision, ICCV'99*, Corfou, vol. 1, p. 262-268, 1999.
- [MAR 96b] MARTINET P., GALLICE J., KHADRAOUI D., «Vision-based control law using 3D visual features », *Proc. World Automation Congress, WAC'96*, Montpellier, vol. 3, p. 497-502, 1996.

- [MAR 97] MARTINET P., DAUCHER N., GALLICE J., DHOME M., «Robot control using monocular pose estimation », *Proc. Workshop on New Trends in Image-based Robot Servoing, IROS'97*, Grenoble, p. 1-12, 1997.
- [MEZ 00] MEZOUAR Y., CHAUMETTE F., «Path planning in image space for robust visual servoing », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'00*, San Francisco, vol. 3, p. 2759-2764, 2000.
- [MIC 93] MICHEL H., RIVES P., Singularities in the determination of the situation of a robot effector from the perspective view of three points. Rapport de recherche INRIA, n° 1850, 1993.
- [MOR 00] MOREL G., LEIBEZEIT T., SZEWCZYK J., BOUDET S., POT J. «Explicit incorporation of 2D constraints in vision-based control of robot manipulators », *Proc. Int. Symp. on Experimental Robotics, Sydney, LNCIS*, vol. 250, p. 99-108, Springer Verlag, 2000.
- [MOT 92] MOTYL G., Couplage d'une caméra et d'un faisceau laser en commande référencée vision, Thèse de l'Université Blaise Pascal de Clermont-Ferrand, LASMEA, 1992.
- [NAY 96] NAYAR S.K., NEME S.A., MURASE H., «Subspace methods for robot vision », *IEEE Trans. on Robotics and Automation*, vol. 12, n° 5, p. 750-758, 1996.
- [NEL 94a] NELSON B., KHOSLA P.K., «Integrating sensor placement and visual tracking strategies », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'94*, p. 1351-1356, 1994
- [NEL 94b] NELSON B., KHOSLA P.K., «Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance », *Int. Journal of Robotics Research*, vol. 14, n° 3, p. 255-269, 1995.
- [NEL 95] NELSON B., KHOSLA P.K., «The resolvability ellipsoid for visual servoing », *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, Seattle, p. 829-832, 1994.
- [PAP 93] PAPANIKOLOPOULOS N., KHOSLA P.K., 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, p. 14-35, 1993.
- [PAP 95] PAPANIKOLOPOULOS N., «Selection of features and evaluation of visual measurements during robotic visual servoing tasks », *Journal of Intelligent and Robotics Systems*, vol. 13, p. 279-304, 1995.
- [PIS 93] PISSARD-GIBOLLET R., Conception et commande par asservissement visuel d'un robot mobile, Thèse de l'Ecole des Mines de Paris, INRIA Sophia-Antipolis, 1993.
- [PHO 95] PHONG T.Q., HORAUD R., YASSINE A., TAO P.D., «Object pose from 2D to 3D point and line correspondences », *Int. Journal of Computer Vision*, vol. 15, n° 3, p. 225-243, 1995.
- [REM 98] RÉMY S., Etalonnage d'un système de vision embarquée, Thèse de l'Université Blaise Pascal de Clermont-Ferrand, LASMEA, 1998.
- [REY 98] REYES F., KELLY R., Experimental evaluation of fixed-camera direct visual controllers on a direct-drive robot. *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'98*, Louvain, vol. 3, p. 2337-2657, 1998.
- [RIV 01a] RIVES P., DEVY M., «Perception pour la localisation », In *La robotique mobile*, p. 141-198, Traité IC2, Hermès, Paris, 2001.
- [RIV 01b] RIVES P., DEVY M., «Perception pour la navigation et la commande », In *La robotique mobile*, p. 199-257, Traité IC2, Hermès, Paris, 2001.

- [RIZ 96] RIZZI A., KODITSCHKEK D., «An active visual estimator for dexterous manipulation », *IEEE Trans. on Robotics and Automation*, vol. 12, n° 5, p/ 697-713, 1996.
- [ROB 93] ROBERT L., Perception stéréoscopique de courbes et de surfaces tridimensionnelles. Application à la robotique mobile, Thèse de l'Ecole Polytechnique, INRIA Sophia Antipolis, 1993.
- [RUF 99] RUF A., HORAUD R., «Rigid and articulated motion seen with an uncalibrated stereo rig », *Proc. Int. Conf. on Computer Vision, ICCV'99*, Corfou, 1999.
- [SAF 92] SAFAEE-RAD R., TCHOUKANOV I., SMITH K.C., BENHABIB B., «Three-dimensional location estimation of circular features for machine vision », *IEEE Trans. on Robotics and Automation*, vol. 8, n° 5, p. 624-639, 1992.
- [SAM 91] SAMSON C., LE BORGNE M., ESPIAU B., *Robot Control: The Task Function Approach*, Clarendon Press, Oxford, 1991.
- [SHA 95] SHARMA R., HUTCHINSON S., «Optimizing hand/eye configuration for visual servo systems », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA '95*, Nagoya, p. 172-177, 1995.
- [SMI 94] SMITH C., PAPANIKOLOPOULOS N., «Computation of shape through controlled active exploration », *IEEE Int. Conf. on Robotics and Automation, ICRA '94*, vol. 3, p. 2516-2521, San Diego, 1994.
- [SUH 93] SUH I.H., «Visual servoing of robot manipulators by fuzzy membership function based neural networks », In [HAS 93a], p. 285-315, 1993.
- [SWA 99] SWAIN OROPEZA R., Contrôle de tâches référencées vision pour la navigation d'un robot mobile en milieu structuré, Thèse de l'Université Paul-Sabatier de Toulouse, LAAS, 1999.
- [TON 97] TONKO M., SCHÄFER K., HEIMES F., NAGEL H.H., «Towards visually servoed manipulation of car engine parts », *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA '97*, vol. 4, p. 3166-3171, Albuquerque, 1997.
- [TSA 89] TSAI R.Y., LENZ R., «A new technique for fully autonomous and efficient 3D robotics hand-eye calibration », *IEEE Trans. on Robotics and Automation*, vol. 5, p. 345-358, 1989.
- [TSA 98] TSAKIRIS D., RIVES P., SAMSON C., «Extending visual servoing techniques to nonholonomic mobile robots », In *The Confluence of Vision and Control, LNCIS*, vol. 237, p. 106-117, Springer Verlag, 1998.
- [VIN 00] VINCZE M., HAGER G., EDS., *Robust Vision for Vision-based Control of Motion*, IEEE Press, Piscataway, 2000.
- [WEI 84] WEISS L.E., Dynamic Visual Servo Control of Robots. An Adaptive Image-based Approach, PhD Thesis, CMU-RI-TR-84-16; Carnegie Mellon University, 1984.
- [WEL 96] WELLS G., VENAILLE C., TORRAS C., «Vision-based robot positioning using neural networks », *Image and Vision Computing*, vol. 14, p. 715-732, 1996.
- [WIL 96] WILSON W., HULLS C., BELL G., «Relative end-effector control using cartesian position-based visual servoing », *IEEE Trans. on Robotics and Automation*, vol. 12, n° 5, p. 684-696, 1996.
- [XIE 89] XIE M., Contribution à la vision dynamique : reconstruction 3D d'objets polyédriques par une caméra mobile, Thèse de l'Université de Rennes 1, IRISA, 1989.

Index

asservissement visuel, 89
asservissement visuel 2D, 116
asservissement visuel 2D 1/2, 114
asservissement visuel 3D, 112

calcul de pose, 102
calibration bras-cil, 93
calibration d'une camra, 96
camra dporte, 93
camra embarque, 92
capteur proximtrique, 126
conditionnement, 119
consigne visuelle, 108

fonction de tche, 108

informations visuelles, 91
informations visuelles 2D, 94
informations visuelles 3D, 102

liaisons virtuelles, 119

matrice d'interaction, 92
matrice de combinaison, 108
moments 2D, 100

paramtres intrinsques, 96
poursuite de cible, 125
primitives gomtriques, 97
projection perspective, 94

surface des limbes, 97

torseur cinmatique, 92
tche hybride, 122

zoom commandable, 97