



**HAL**  
open science

## Localisation référencée modèle d'un robot mobile d'intérieur

Omar Ait Aider

► **To cite this version:**

Omar Ait Aider. Localisation référencée modèle d'un robot mobile d'intérieur. Traitement du signal et de l'image [eess.SP]. Université d'Evry-Val d'Essonne, 2002. Français. NNT : . tel-00682235

**HAL Id: tel-00682235**

**<https://theses.hal.science/tel-00682235>**

Submitted on 23 Mar 2012

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

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

# **THESE**

présentée devant

l'UNIVERSITE D'EVRY – VAL D'ESSONNE

en vue de l'obtention du

**DOCTORAT DE L'UNIVERSITE D'EVRY**

**Spécialité : Robotique**

Par

**Omar AÏT-AIDER**

Localisation référencée modèle d'un robot mobile d'intérieur

Le 19 Décembre 2002

JURY :

J. Droulez (Prés)

E. Colle (Dir.)

M-J. Aldon (Rap.)

M. Mouaddib (Rap.)

M. Mallem

P. Hoppenot



## Résumé

Le présent travail porte sur la localisation incrémentale et absolue d'un robot mobile dans un environnement d'intérieur partiellement modélisé en utilisant la vision monoculaire.

L'environnement de navigation du robot est à base de primitives géométriques (segments). Il intègre la notion d'occlusion grâce à un découpage de l'espace 2-D navigable en Régions d'Invariance Visuelle. Le modèle de caméra à perspective pleine est obtenu grâce au calibrage par la méthode de Zhang.

L'approche adoptée est composée de quatre étapes : acquisition d'une image à partir de la position courante du robot, extraction des primitives observées, mise en correspondance des primitives de l'image avec celles du modèle et calcul de la position et de l'orientation de la caméra.

Deux méthodes numériques de calcul de la position et de l'orientation de la caméra grâce à des correspondances de droites sont présentées et adaptées au cas spécifique de la robotique mobile.

Enfin, un algorithme de mise en correspondance des segments de l'image avec ceux du modèle est défini. Il est basé sur la recherche dans un arbre d'interprétation. Les Régions d'Invariance Visuelle et la configuration du système sont utilisées pour réduire l'espace des correspondances. Des contraintes géométriques d'ordre un et deux sont définies pour assurer l'élagage rapide de l'arbre. Une nouvelle fonction de vérification de la cohérence globale permet de sélectionner l'hypothèse de correspondance la plus cohérente.

Mots clefs : robotique mobile, localisation référencée modèle, calcul de la pose de caméra, mise en correspondance 3-D/2-D.

## Abstract

An incremental and absolute mobile robot self localization method in a partially modelled indoor environment is presented.

A wire frame representation of the environment is adopted. The notion of occlusion is taken into account using View Invariant Regions. A pin-hole model of the camera is obtained thanks to Zhang calibration method.

The localization approach is composed of four steps : image acquisition from robot current position and orientation, image feature extraction, 3-D/2-D feature matching and camera pose recovery.

Two full perspective camera pose recovery methods using straight line correspondances and numerical optimisation technic are presented. Adaptation of these methods to mobile robotics context is defined.

Finally, the crucial problem of matching image features to model features is achieved using an algorithm based on Interpretation Tree Search. Dimension of correspondance space is reduced using View Invariant Regions and the specific configuration of the robot. Two geometric constraints are used to efficiently prune the Interpretation Tree testing the local consistency. A new function is defined to test the global consistency and to select the best matching hypothesis.

Keywords : mobile robot localization, map-based localization, camera pose recovery, feature matching.

# Table des matières

<b>Résumé</b> .....	<b>3</b>
<b>Abstract</b> .....	<b>3</b>
<b>Table des matières</b> .....	<b>4</b>
<b>Liste des figures</b> .....	<b>5</b>
<b>Liste des tables</b> .....	<b>7</b>
<b>Introduction générale</b> .....	<b>8</b>
<b>Localisation en robotique mobile</b> .....	<b>12</b>
Localisation en robotique mobile : techniques et capteurs.....	13
Localisation en environnement intérieur .....	19
Localisation basée sur le modèle grâce à la vision.....	22
Conclusions et directions de recherche .....	25
<b>Outils pour la localisation par vision monoculaire</b> .....	<b>28</b>
Modélisation et calibrage du capteur.....	29
Modélisation de l'environnement.....	43
Extraction des primitives de l'image.....	48
<b>Calcul de la position et de l'orientation de la caméra</b> .....	<b>56</b>
Etat de l'art .....	57
Méthodes développées.....	70
Evaluation des méthodes développées .....	82
Conclusions .....	90
<b>Mise en correspondance 3D-2D</b> .....	<b>91</b>
Etat de l'art .....	92
Méthode développée.....	102
Evaluation de la méthode développée .....	123
Conclusions .....	127
<b>Annexe A</b> .....	<b>131</b>
<b>Annexe B</b> .....	<b>135</b>
<b>Annexe C</b> .....	<b>137</b>
<b>Références</b> .....	<b>138</b>

# Liste des figures

Figure 1.1 : formalisme du problème de localisation d'un robot mobile	13
Figure 1.2 : calcul de la position grâce à l'odométrie ([Hoppenot97])	14
Figure 1.3 : la trilatération	17
Figure 1.4 : principe de la triangulation	17
Figure 1.5 : calcul de la position à partir de données goniométriques	18
Figure 1.6 : localisation grâce à deux points	19
Figure 1.7: exemple de représentation topologique d'un environnement constitué de couloir (C) et de pièces (P)	21
Figure 1.8 : approche de localisation combinant l'odométrie et la vision	27
Figure 2.1 : passage du repère de la scène à l'image	33
Figure 2.2 : les trois différents modèles de projection	35
Figure 2.3 : images utilisées pour le calibrage	41
Figure 2.4 : exemple de modèle fil de fer d'un environnement intérieur	44
Figure 2.5 : construction de la frontière d'une région d'invariance visuelle	46
Figure 2.6 : limitation de l'angle de vue d'un segment vertical $L$ du modèle dans une $RIV$	47
Figure 2.7 : limitation de l'angle de vue d'un segment horizontal $L$ du modèle dans une $RIV$	47
Figure 2.8 : détection de contours par l'approche gradient et l'approche Laplacien	49
Figure 2.9 : représentation d'une droite dans l'espace de Hough	51
Figure 2.10 : table de Hough	51
Figure 2.11 : chaînage des points de contours	52
Figure 2.12 : résultat de l'extraction des primitives de la scène	54
Figure 3.1 : méthode analytique à trois points	60
Figure 3.2 : méthode analytique à quatre trois points coplanaires	61
Figure 3.3: projection d'un segment 3D	63
Figure 3.4 : repère-caméra, repère-image et repère-modèle	64
Figure 3.5 : utilisation des appariements de segments de droites dans la méthode de Lowe	74
Figure 3.6 : degrés de liberté de la caméra en robotique mobile	75
Figure 3.7 : projection d'un segment 3D	78
Figure 3.8: résultats de la localisation des quatre méthodes en terme de précision de localisation	87
Figure 3.9 : performance de la localisation en fonction de la qualité d'initialisation	88
Figure 3.10 : évolution de l'erreur de localisation en fonction du nombre d'appariements utilisés	89
Figure 3.11 : modèle ayant servi à l'évaluation des méthodes de calcul de la position et de l'orientation sur des images réelles.	90
Figure 3.12.a : erreur de localisation pour $Q_i=1$	90
Figure 3.12.b : erreur de localisation pour $Q_i=4$	90
Figure 4.1 : construction de l'arbre d'interprétation	96
Figure 4.2 : exemple de zone d'erreur pour des primitives segments	97
Figure 4.3 : principe de recherche d'appariements dans le système FINALE	102
Figure 4.4 : contrainte unaire	108

Figure 4.5 : contrainte binaire	109
Figure 4.6 : contrainte binaire	110
Figure 4.7 : évolution de la pente du segment de projection d'un segment vertical dans le modèle en fonction de l'angle $\theta$ de la caméra.	111
Figure 4.8 : contrainte binaire dans le cas général	111
Figure 4.9 : seules les lignes en gras peuvent correspondre à des segments verticaux	113
Figure 4.10 : notion de distance entre deux segments dans le plan image	115
Figure 4.11 : schéma synoptique de l'algorithme de mise en correspondance	117
Figure 4.12 : propagation de l'incertitude des paramètres des primitive de l'image sur la zone de contrainte de la position avec la contrainte unaire.	122
Figure 4.13 : propagation de l'incertitude des paramètres des primitive de l'image sur la zone de contrainte de la position avec la contrainte binaire.	124
Figure 4.14 : prise en compte du bruit sur les paramètres du cercle de contrainte de la position	124
Figure 4.15 : performances de l'élagage grâce aux contraintes unaire et binaire avec trois différents niveaux de qualité de l'estimation initiale de la position et de l'orientation.	128
Figure 4.16 : modèle simple utilisé dans les tests de mise en correspondance (vue du pattern utilisé à gauche et ensemble de segments composant le modèle à droite)	130
Figure 4.17 : a-environnement utilisé pour les test, b-modèle correspondant, c et e-exemples de scènes segmentées, d et f-exemples de résultats d'appariements.	131

# Liste des tables

Table 2.1 : erreur de reconstruction 2D sur des points n'ayant pas participé au calibrage.	44
Table 3.1 : dérivées partielles de $u$ et $v$ par rapport aux paramètres de position et d'orientation	72
Table 3.2 : dérivées partielles de $u$ et $v$ par Carceroni	73
Table 3.3 : valeurs des paramètres de bruit utilisés dans le test 1	85
Table 3.4 : valeurs des paramètres concernant les résultats du test1	86
Table 3.5 : initialisations utilisées pour le test 2	87
Table 4.1 : valeurs de $Q_i$ utilisées dans les tests	26
Table 4.2 : résultats de l'algorithme d'appariement	127
Table 4.2 : échantillons de résultats de mise en correspondances sur des images réelles	130



## *Introduction générale*

Les robots mobiles sont largement utilisés dans les environnements industriels pour le transport de produits par exemple. Le plus souvent ces tâches sont répétitives et suivent un chemin bien défini, parfois même bien matérialisé comme des lignes sur le sol ou des amers artificiels.

Il y a actuellement une forte tendance à élargir les milieux où évoluent les robots à des environnements de bureaux ou à des environnements domestiques (robots de service). Les types d'applications possibles sont innombrables. Cela peut être des tâches de nettoyage et d'entretien ou encore une assistance à une personne handicapée dans des tâches d'exploration et de préhension. Un robot peut également servir de guide pour la visite d'un musée. On parle alors, de façon générale, de robotique d'intérieur.

Un tel cadre d'utilisation requiert que le système robotisé dispose d'un niveau minimum d'autonomie et de facilités de navigation. Pour ce faire, le système doit généralement accomplir trois tâches de base qui sont la localisation, la planification et la navigation.

Parmi ces tâches, la localisation relativement à l'environnement occupe une place de choix puisqu'elle détermine le bon déroulement des deux autres. Elle consiste à calculer et à maintenir à jour la connaissance de la position et de l'orientation du robot dans un repère absolu lié à l'environnement de travail. On se restreint ici au cas de systèmes navigant sur un plan. Le robot est alors complètement localisé par deux paramètres de position et un paramètre d'orientation.

S'il est permis d'équiper un environnement industriel de marqueurs ou balises artificielles pour assurer le maintien de la connaissance de la position du robot, ceci n'est pas le cas dans un environnement domestique ou des bureaux administratifs. Dans ce cas il est très désirable que la localisation n'ait pas recours à la modification de l'environnement.

Parmi les différentes approches de localisation absolue existantes, une technique très répandue est celle basée sur un modèle (map-matching ou model-based localisation). En effet la localisation dans un environnement intérieur utilise le plus souvent une carte, ou modèle, de l'environnement. Le principe général est que le robot perçoit en ligne et grâce à ces capteurs extéroceptifs un modèle local de son espace de travail qu'il met en rapport avec le modèle global puis estime la position et l'orientation courantes à partir desquelles il a acquis ces données.

Le modèle peut se présenter sous diverses formes comme les grilles d'occupation, les modèles à primitives géométriques ou encore les modèles topologiques. Dans un environnement construit par l'homme, il est assez naturel de représenter l'environnement par un ensemble de primitives géométriques. Le type de primitives utilisées dépend entre autre du type de capteur. En effet, les informations issues du capteur et celles stockées dans le modèle doivent pouvoir être comparées.

Quel que soit le capteur utilisé, la localisation basée sur un modèle passe par quatre étapes principales qui sont : l'acquisition de données capteur brutes, l'extraction d'informations pertinentes, la mise en correspondance des données observées avec celles du modèle et l'estimation de la position et de l'orientation.

Cette succession de plusieurs traitements, qui généralement sont complexes ou mettent en jeu de grandes quantités de données, rend les méthodes basées sur le modèle coûteuses en temps

de calcul. Les problèmes de mise en œuvre en ligne sont donc un aspect majeur du problème de localisation. La plupart des systèmes adoptent une approche qui consiste à combiner ces calculs avec une localisation relative (généralement odométrique). Cette dernière permet de maintenir à jour la connaissance de la position et de l'orientation avec un très haut niveau d'échantillonnage mais sa précision se détériore au fil du déplacement à cause de l'accumulation d'erreurs. Ce système permet de réduire la périodicité du recours à la localisation absolue. De plus, il lui fournit une estimation de la position et de l'orientation pour réduire la complexité et le temps de calcul. On parle alors de localisation incrémentale (ou tracking). Cependant, il arrive que cette estimation ne soit pas disponible (comme lors de la mise en service ou s'il y a défaillance de l'odométrie). Dans ce cas la localisation est dite globale.

Les problèmes de mise en œuvre en ligne sont d'autant plus ardues lorsque l'on utilise la vision par ordinateur. Dans ce cas, les données acquises sont des images issues d'une caméra embarquée. L'utilisation de la vision en robotique mobile a toujours séduit les chercheurs du domaine à cause de l'analogie que l'on peut faire avec le système de localisation humain. De plus ce capteur est l'un des plus riches en quantité d'informations véhiculées. En revanche, son utilisation implique un certain nombre de problèmes. Premièrement, une image contient un très grande quantité de données brutes dont seule une partie comporte des informations pertinentes. Dans le cadre d'une application en robotique, il est indispensable de pouvoir identifier les zones d'intérêt de l'image et de les représenter sous forme d'informations concises. Deuxièmement, l'image représente un environnement 3D sous forme de données 2D. La reconstruction de la notion de profondeur nécessite alors une connaissance pointue de la nature de la transformation reliant les deux domaines.

En supposant que le modèle interne de la caméra est connu et que le modèle de l'environnement est disponible, la localisation grâce à la vision monoculaire suit le même schéma général composé des quatre étapes citées plus haut :

- Acquisition de l'information capteur. Le robot saisit une image de son environnement à partir de sa position courante.
- Détection de primitives. Cette étape correspond généralement aux opérations de traitement d'image telles que l'extraction de contours et la segmentation.
- Mise en correspondance des observations avec le modèle : Vise à trouver pour des appariements entre les primitives de l'image et celles du modèle si ces appariements existent.
- Calcul de la position et de l'orientation de la caméra (donc du robot). Se fait grâce aux appariements établis dans l'étape précédente.

La performance du système de localisation final en terme de temps de calcul et de précision dépend donc de celle de chacune des étapes de calcul en ligne mais aussi des choix effectués quant au modèles de la caméra et celui de l'environnement.

Le présent travail a pour but la mise en œuvre d'un système de localisation incrémentale basé sur la vision monoculaire. Le modèle de l'environnement est construit hors ligne. Les paramètres intrinsèques de la caméra sont également obtenus par calibrage hors ligne. L'objectif du travail a été d'optimiser chacune des étapes de traitement pour qu'elles répondent au mieux aux exigences du contexte. Les principaux apports sont d'une part l'adaptation de méthodes numériques de calcul de la position et de l'orientation de la caméra au contexte de localisation d'un robot mobile et d'autre part le développement d'une stratégie efficace de mise en correspondance des primitives de l'image avec celles du modèle.

Le manuscrit est organisé comme suit :

Le Chapitre 1 est une introduction à la problématique de la localisation en robotique mobile. Les techniques les plus utilisées sont répertoriées. La localisation basée sur le modèle y est abordée. En fin de chapitre, les principaux aspects de la localisation grâce à la vision monoculaire sont introduits.

Le Chapitre 2 regroupe les outils de base nécessaires à la mise en œuvre d'un système de localisation basé sur la vision monoculaire. La première partie concerne le calibrage du capteur. Elle commence par une étude des différentes techniques existantes. Les paramètres intrinsèques devant être valides dans un volume important autour de la caméra (comparativement à une tâche de bras manipulateur par exemple), une méthode bien appropriée à notre contexte, la méthode de Zhang, a été retenue. La seconde partie est consacrée à la construction du modèle de l'environnement. Le type de primitives choisi est le segment de droite. Ce choix est motivé par la facilité de détection de cette primitive dans l'image et de la robustesse de ses paramètres calculés en présence de bruit. Un modèle filaire associé à une carte 2D de l'espace navigable réparti en régions d'invariance visuelle (RIV) est établi. La troisième partie porte sur l'extraction des primitives de l'image. Une méthode simple de détection et d'approximation polygonale des segments est présentée.

Le Chapitre 3 est consacré au calcul de la position et de l'orientation de la caméra grâce à des correspondances de droite et en utilisant des techniques d'optimisation numérique. Cette étape est présentée avant celle de la mise en correspondance pour faciliter la compréhension de cette dernière. En effet, le processus composé des quatre étapes citées plus haut n'est pas linéaire. Il alterne recherche de correspondances 2D-3D et calcul de la position et de l'orientation. On suppose donc dans ce chapitre que la bonne combinaison d'appariements entre les primitives de l'image et celles du modèle est connue. Le chapitre commence par une étude des différentes techniques existantes. Ensuite, deux méthodes développées, une basée sur l'approche de Lowe et l'autre sur l'approche de Phong-Horaud, sont présentées. Ces approches, généralement utilisées pour la localisation 3-D dans le domaine de la reconnaissance d'objets, ont été adaptées au contexte de la robotique mobile. Elles utilisent un modèle perspectif de caméra. Une analyse de leurs performances en terme de sensibilité par rapport au bruit, aux conditions initiales et au nombre d'appariements disponibles mais aussi l'existence de singularités est présentée. Une évaluation expérimentale comparative effectuée avec des données synthétiques et réelles termine le chapitre.

Le Chapitre 4 porte sur le problème de la mise en correspondance des segments de l'image avec ceux du modèle. C'est là le point le plus important de la plupart des applications basées sur la vision. Pour situer la problématique et les principales difficultés inhérentes à cette tâche, les principales approches de mise en correspondance sont présentées. Après justification des choix arrêtés en accord avec le contexte, l'algorithme de mise en correspondance développé dans le cadre de ce travail est présenté. La stratégie adoptée est basée sur la recherche arborescente. Des contraintes d'ordre un et deux sont définies. Elles prennent en compte le modèle perspectif de caméra et sont utilisables directement sur des appariements 2D-3D. Elles sont utilisées pour assurer un élagage efficace de l'arbre d'interprétation. La prise en compte de l'incertitude due au bruit des données image est également abordée. Une évaluation expérimentale sur des images synthétiques et réelles est présentée en fin de chapitre.



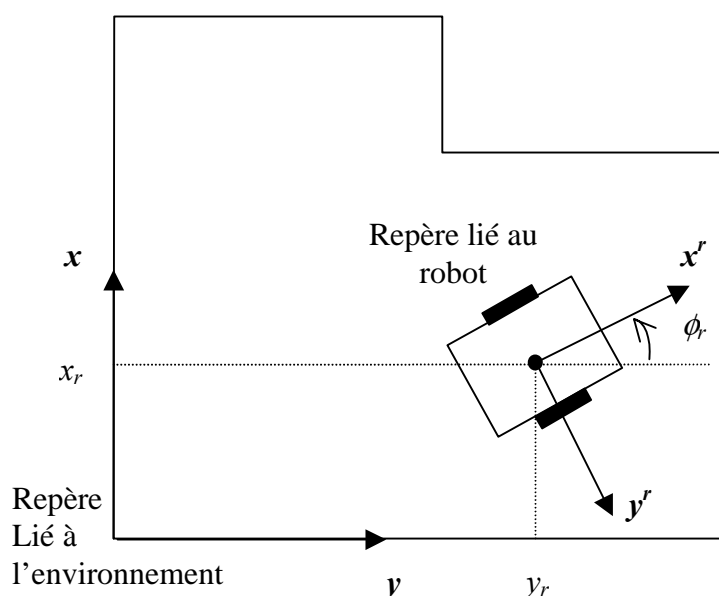
# **Localisation en robotique mobile**

Ce chapitre est une introduction au domaine de la localisation en robotique mobile. La problématique de la localisation et l'intérêt de la connaissance de la position et de l'orientation du robot par rapport à son environnement y sont présentés. Les différentes approches de localisation ainsi que les capteurs associés sont énoncés. La localisation en environnement intérieur est ensuite abordée. Le cas particulier de la localisation basée sur un modèle grâce à la vision est introduit. Les problèmes à résoudre dans ce cas sont situés. En fin de chapitre les objectifs du travail effectué sont soulignés.

## Localisation en robotique mobile : techniques et capteurs

Quelque soit le domaine d'application pour lequel il est destiné, un robot mobile, pour être utilisable, doit comporter un système permettant un certain niveau d'autonomie dans la localisation et la navigation. Pour schématiser, il doit être capable de répondre à trois types de questions : « où suis-je ? », « où vais-je ? » et « comment y aller ? ». La première question soulève le problème de la localisation. Les deux autres sont liées à la planification de trajectoire et à la navigation proprement dite. La bonne exécution des deux dernières tâches est fortement liée à la première.

Cette étude se restreint aux robots mobiles navigant sur un plan (2-D), ce qui englobe une très large partie des systèmes existants. Localiser le robot revient alors à déterminer trois paramètres : deux coordonnées cartésiennes pour la position et un angle pour l'orientation. De façon plus formelle, la tâche de localisation consiste à calculer la transformation de passage d'un repère lié au robot à un repère lié à l'environnement (figure 1.1).



*Figure 1.1 : formalisme du problème de localisation d'un robot mobile*

Plusieurs techniques et méthodes ont été développées pour assurer la connaissance exacte et de façon autonome de la position d'un robot mobile dans son environnement. A ce jour, ces techniques peuvent être regroupées en deux catégories principales : les méthodes de localisation relatives et les méthodes de localisation absolue.

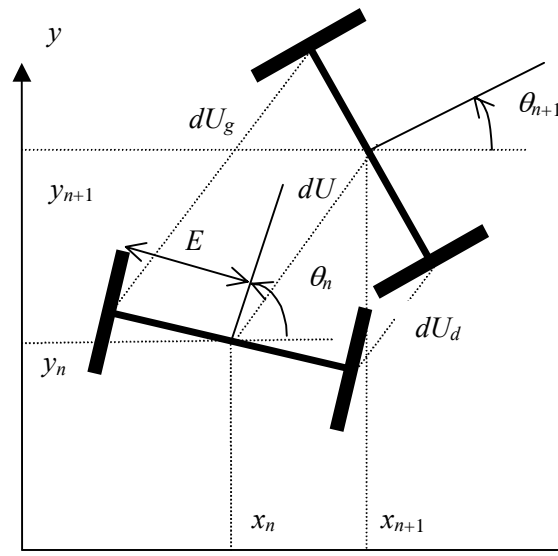
### Localisation relative

La position du robot est calculée en incrémentant sa position précédente de la variation mesurée grâce à des capteurs proprioceptifs. Les deux principales méthodes de localisation relative sont la localisation grâce à l'odométrie et la localisation inertielle.

#### Odométrie

C'est l'un des systèmes les plus utilisés en robotique mobile car il présente beaucoup d'avantages comme le coût financier, un très haut niveau d'échantillonnage de la mesure, une très bonne précision à court terme et une très grande facilité de mise en oeuvre

([Cox91],[hoppenot97],[Borenstein97]). L'idée fondamentale de ce système est l'intégration de l'incrément de la position, calculé grâce à des encodeurs montés sur les roues, par rapport



**Figure 1.2 :** calcul de la position grâce à l'odométrie ([Hoppenot97])

au temps. Les déplacements  $dU_g$  et  $dU_d$  des roues droite et gauche permettent de calculer la variation de l'orientation ainsi que la variation de la position  $dU$  entre l'état  $n$  et l'état  $n+1$  (figure 1.2). Il en résulte malheureusement une accumulation non bornée de l'erreur. L'erreur en orientation induit d'importantes erreurs en position et les deux erreurs croissent en fonction de la distance parcourue. Les erreurs peuvent être regroupées en deux catégories selon leur source. Les erreurs systématiques résultent des imperfections du modèle géométrique du robot (diamètres des roues différents, incertitude sur les dimensions des axes de la base,...). Les erreurs non systématiques résultent de l'interaction entre le robot et son environnement telle que les glissements ou les chocs qui ne sont pas pris en compte dans la mesure du mouvement effectué.

## Localisation inertielle

Cette technique utilise des accéléromètres pour calculer l'accélération subie par la base mobile et des gyroscopes pour calculer la variation de sa rotation [Barshan95]. L'intégration de cette mesure (ou la double intégration dans le cas de l'accéléromètre) permet de calculer la variation de la position. Les capteurs utilisés dans ce type de localisation présentent l'avantage d'être « auto-suffisants » puisqu'ils ne nécessitent aucune référence externe. Cependant, une erreur même minime est amplifiée par l'intégration. De plus, le rapport signal/bruit n'est pas très élevé. Dans le cas des accéléromètres, l'environnement de travail doit être parfaitement horizontal sous peine de détecter une composante de la gravitation terrestre comme étant due à un déplacement du robot.

## Localisation absolue

La position est calculée par rapport à des repères fixes grâce à des capteurs extéroceptifs. Ceci requiert souvent la connaissance de l'environnement. Les différentes techniques peuvent être distinguées par la nature des repères utilisés ou par la méthode de calcul. Selon la nature des repères utilisés, les approches les plus connues sont la localisation grâce aux compas magnétiques, la localisation grâce à des repères actifs, la localisation grâce à des repères

passifs et la localisation basée sur le modèle. Selon les techniques de calcul utilisées on distingue, entre autres, les méthodes basées sur la trilatération (ou multilatération), les méthodes basées sur la triangulation.

## **Types de repères utilisés**

### **Localisation par compas magnétique**

Un compas magnétique permet de déterminer une orientation absolue en mesurant la composante horizontale du champ magnétique terrestre. Le repère par rapport auquel on se positionne est dans ce cas lié à la terre. L'inconvénient principal est le fait que ce champ soit altéré au voisinage de lignes à haute tension ou par les grandes structures métalliques. Un exemple d'application est présenté dans [Byrne92].

### **Localisation par balises actives**

Des balises sont disposées à des emplacements connus de l'environnement de travail. Elles sont facilement détectées par le robot et avec un faible coût calculatoire. La base mobile à localiser est dotée d'un émetteur et les balises de récepteurs ou inversement. L'avantage d'un tel système est le taux d'échantillonnage élevé. Les inconvénients sont la difficulté à disposer les balises avec précision et le coût d'installation et de maintenance. Le calcul de la position et de l'orientation est basé sur la trilatération ou la triangulation.

### **Localisation grâce à des repères passifs**

On distingue deux types de repères passifs, les balises passives et les amers. Contrairement aux balises actives, les balises passives se contentent de réfléchir un signal provenant de l'équipement de mesure. Il peut s'agir par exemple de miroirs ou de catadioptré. Ces balises peuvent également comporter des informations plus évoluées (codes-barres). Elles doivent avoir une position connue dans l'environnement et doivent être facilement identifiables grâce à un contraste suffisant par rapport au « fond ».

On appelle amers des éléments distincts de l'environnement que le robot peut reconnaître aisément grâce à ses capteurs extéroceptifs. Ces amers peuvent être par exemple des formes géométriques (rectangles, lignes, cercles). Ce sont des éléments déjà existant dans l'environnement et qui ont une fonction autre que celle de localiser le robot. Ils présentent l'avantage de ne pas modifier l'environnement de travail ce qui est important dans une application comme l'assistance à des personnes dans un appartement par exemple. Cependant, leur détection et leur identification peuvent être plus difficiles et les risques d'ambiguïté ou de fausses détections plus élevés. Le plus souvent, le capteur utilisé est visuel ([Talluri93]). L'avantage de ce type de repère par rapport aux balises actives est le faible coût. Leur inconvénients sont la portée réduite et la difficulté d'identification.

### **Localisation basée sur un modèle**

Dans cette technique, un plan ou modèle de l'espace de travail du robot est stocké en mémoire. A un instant donné, le robot perçoit et construit grâce à ces capteurs un plan local de son environnement. Le système effectue alors une mise en correspondance entre le plan local et la partie correspondante dans le modèle global. Si une telle mise en correspondance est trouvée, alors la position et l'orientation du robot peuvent être calculées.

Il existe plusieurs façons de représenter l'environnement. Dans [Thrun96] deux grandes représentations sont distinguées. Il s'agit des grilles d'occupation et des cartes topologiques. Cependant cette distinction ne fait pas apparaître les modèles à primitives géométriques

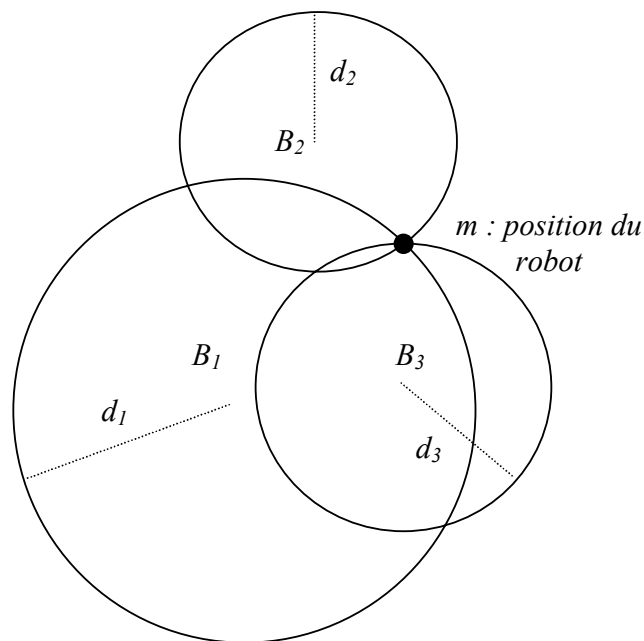


([Moutarlier90, [Leonard91],...). On peut donc regrouper les façons de représenter l'environnement en deux grandes familles : les représentations métrique avec un ensemble de primitives géométriques ou avec des grilles d'occupation composées de cellules et les représentations topologiques (non métriques). Le modèle peut être pré-établi et stocké en mémoire dans certains cas. Dans d'autres cas il peut être inexistant au départ. C'est le robot qui le construit grâce à ses capteurs par exploration (SLAM ou localisation et modélisation simultanée). Nous reviendrons plus en détail sur ce type de localisation dans la section suivante.

## Méthodes de calcul de la position et de l'orientation

### La trilatération (ou multilatération)

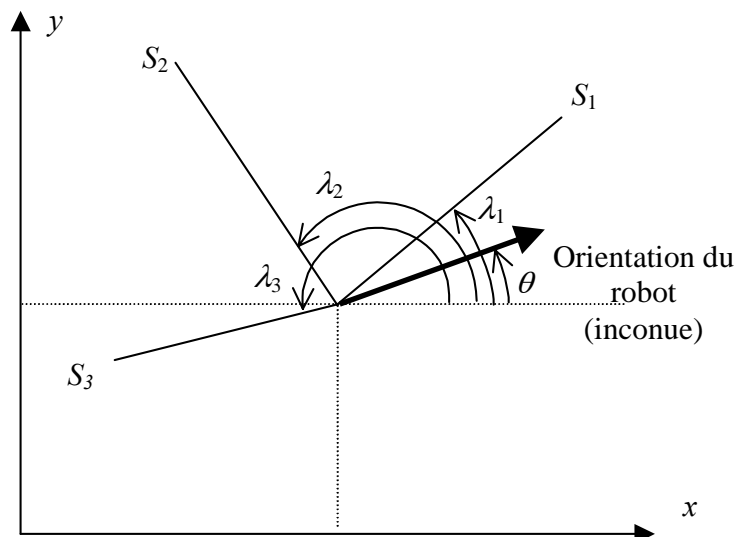
Le calcul de la position se fait grâce à la connaissance de la distance de la base à trois références de l'environnement au moins. Le temps de vol de l'onde émise peut par exemple être utilisée pour le calcul de ces distances. La figure 1.3 montre un exemple de calcul de la position du robot grâce à la connaissance des distances respectives aux balises  $B_1$ ,  $B_2$  et  $B_3$ .



**Figure 1.3 : la trilatération**

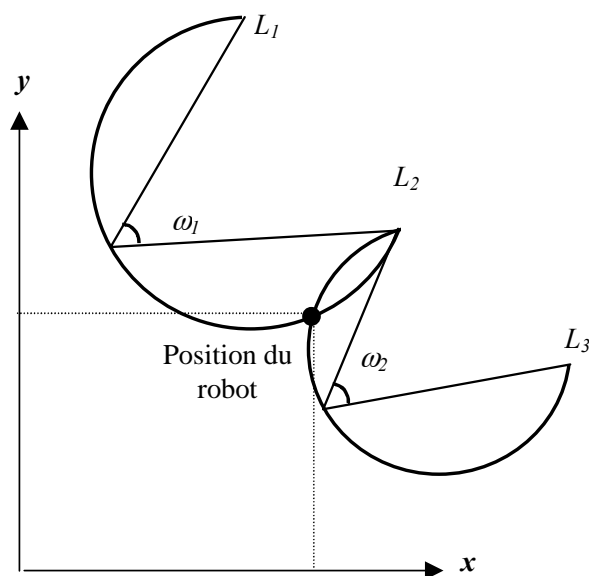
### La triangulation

Dans ce cas les mesures utilisées sont goniométriques. Les angles  $\lambda_1, \lambda_2, \lambda_3, \dots$  avec lesquels sont vus les repères sont calculés (figure 1.4). Ces mesures peuvent par exemple être effectuées grâce à un capteur (récepteur ou émetteur) rotatif embarqué sur le robot ou encore grâce à une caméra panoramique.



**Figure 1.4 :** Principe de la triangulation

Un exemple de calcul de la position à partir de ces angles avec un minimum de trois mesures est représenté en figure 1.5. Chaque couple de mesures contraint la position du robot à appartenir à un arc de cercle défini grâce à la différence  $\omega$  des angles de vue de chaque amer. La position finale retenue correspond à l'intersection de tous les arcs de cercles calculés grâce à tous les couples possibles de mesures.



**Figure 1.5 :** Calcul de la position à partir de données goniométriques

Un algorithme de calcul de la pose par triangulation est présenté dans [Betke98]. Il est basé sur une représentation des amers par des nombres complexes ce qui permet d'obtenir un système d'équations linéaires. La complexité de l'algorithme est linéairement proportionnelle au nombre de mesures.

### Localisation grâce à deux points

Une autre technique de calcul est citée dans [Hoppenot97]. Une caméra et un plan laser sont utilisés. Une coupe de l'environnement est effectuée grâce au laser. Deux points  $R(x_r, y_r)$  et  $L(x_l, y_l)$  dans le repère du robot  $\mathbf{R}_r$ , sont extraits de cette coupe. Un repère  $\mathbf{R}_{ij}$  est défini grâce à ces deux points (figure 1.6). La matrice de passage de  $\mathbf{R}_{ij}$  à  $\mathbf{R}_r$  s'écrit :

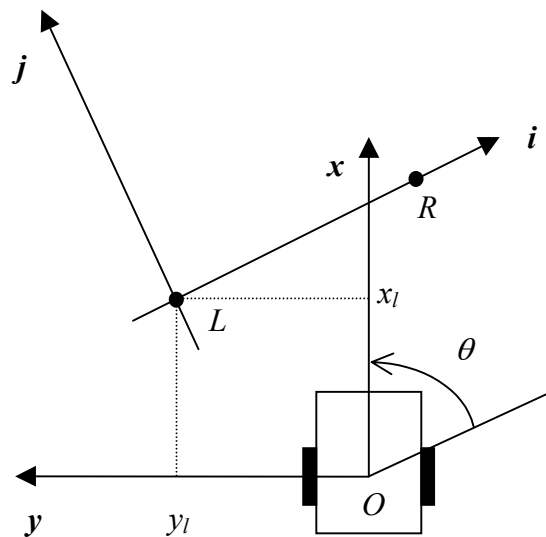
$${}^i j T_r = \begin{bmatrix} \cos \theta & \sin \theta & 0 & -x_l \cos \theta - y_l \sin \theta \\ -\sin \theta & \cos \theta & 0 & x_l \cos \theta - y_l \sin \theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

l'orientation et la position sont alors obtenues comme suit :

$$\begin{cases} \theta = \arctan\left(\frac{y_r - y_l}{x_r - x_l}\right) & \text{si } x_l < x_r \\ \theta = -\frac{\pi}{2} & \text{si } x_l = x_r \\ \theta = \arctan\left(\frac{y_r - y_l}{x_r - x_l}\right) - \pi & \text{si } x_l > x_r \end{cases}$$

et

$$\begin{cases} i = -x_l \cos \theta - y_l \sin \theta \\ j = x_l \cos \theta - y_l \sin \theta \end{cases}$$



**Figure 1.6 : localisation grâce à deux points**

## Localisation en environnement intérieur

Nous nous intéressons à présent au problème de localisation dans un environnement intérieur. Ce problème est apparu suite au développement de projets de robots mobiles connus sous le nom de robots de service. Les premiers robots mobiles étaient appelés AGVs (Automatic Guided Vehicles). Ce sont des systèmes destinés presque exclusivement à des tâches industrielles répétitives telles que le transport de charges dans un environnement structuré en suivant un chemin unique. Cette tâche était exécutée grâce à des bandes réfléchissantes, à une ligne blanche tracée sur le sol ou grâce à des câbles électromagnétiques enterrés. Depuis, les robots mobiles sont de plus en plus développés pour des applications telles que l'assistance à une personne handicapée en appartement ([Hoppenot01]), le nettoyage et l'entretien, ou encore le guidage d'une visite dans un musée. Ces applications ont en commun un environnement que l'on peut qualifier de semi-structuré.

S'il est permis d'équiper un environnement industriel par des balises ou des amers artificiels, ceci est difficilement envisageable dans un environnement domestique ou des bureaux. Il paraît clairement que la localisation basée sur un plan ou un modèle et, dans une moindre mesure, les méthodes utilisant des amers naturels soient les mieux adaptées à ce type de localisation. Il faut noter que dans ce contexte, la limite entre les deux techniques est assez floue. Un aspect important de la localisation en intérieur est la manière de représenter l'environnement, autrement dit, la structure du modèle la plus adéquate en terme de richesse d'information et de concision. A la lumière des travaux étudiés, deux grandes représentations émergent. Il s'agit des modèles topologiques et des modèles métriques (à primitives géométriques ou avec des grilles d'occupation).

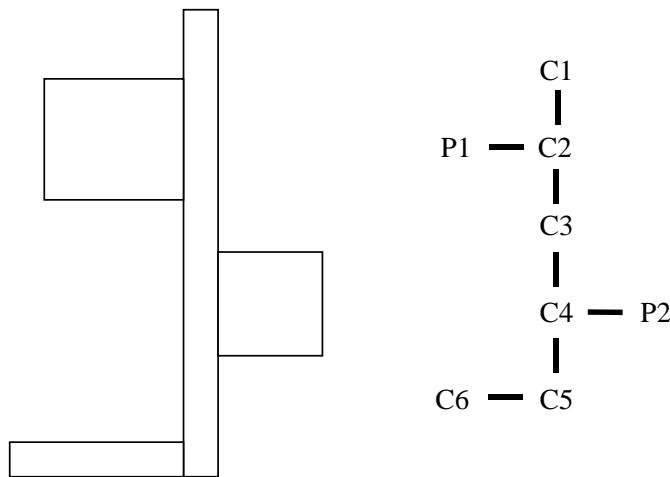
L'approche de localisation et le type du modèle sont fortement liés. Quelque soit l'approche utilisée, la localisation dans le contexte d'un environnement interne peut être divisée en deux problèmes que l'on désignera par localisation incrémentale et localisation globale. La localisation incrémentale suppose l'existence à tout moment d'une estimation même grossière de la position et de l'orientation du robot ce qui permet de faciliter la tâche de mise en correspondance avec le modèle. Dans certains cas aucune estimation n'est disponible. Le problème de localisation, dit global, est alors plus complexe à cause des symétries et des ambiguïtés.

Dans les sections suivantes nous présentons les trois principales représentations de l'environnement ainsi que les approches de localisation associées à chacune.

### Modèle topologique

L'environnement est représenté par un graphe constitué de places reliées par des connecteurs. Selon la définition donnée dans [Chatila85], une place est un espace qui peut être soit topologique soit fonctionnel. Par exemple, une pièce ou un corridor sont des unités topologiques, une imprimante peut représenter une unité fonctionnelle. Des exemples de connecteurs entre les places sont les portes ou les escaliers. L'avantage d'une telle représentation est la concision et la facilité de construction d'un modèle puisque elle ne nécessite pas une connaissance très pointue de l'environnement.

Pour se localiser, le robot a seulement besoin de savoir sur quel connecteur il se déplace en quittant une place. Cependant il doit être capable de distinguer facilement les places. Dans [Ulrich00], le capteur utilisé est une caméra. Des images couleur sont prises à une fréquence de 1Hz et libellées avec l'endroit où elles ont été prises. Les vues sont comparées grâce à des histogrammes à une dimension. Un exemple de modèle topologique est donné en figure 1.7.



*Figure 1.7: exemple de représentation topologique d'un environnement constitué de couloir (C) et de pièces (P)*

## Modèle à primitives géométriques

L'idée de départ est qu'un environnement semi-structuré est constitué de primitives de base détectables par les capteurs extéroceptifs tels que les caméras, les capteurs à ultrasons ou les télémètres laser. Les exemples de primitives les plus courants sont les lignes, les coins, les plans ou les points. Ces primitives ont des positions et des dispositions connues et peuvent être paramétrées par des attributs tels que la longueur, la couleur, la texture, etc.

On peut citer plusieurs travaux utilisant des modèles à primitives géométriques. Dans [Chatila85] le modèle de l'environnement est un plan polygonal. Un capteur laser est utilisé pour détecter des lignes et réaliser un appariement avec le modèle. Un principe similaire est utilisé dans [Hoppenot98] avec des capteurs à ultrasons. Dans [Kuc87] les arêtes et coins de l'environnement détectés grâce à un sonar sont considérés comme amers pour la localisation. Une telle représentation présente deux avantages. D'une part, l'utilisation de ce type de primitives permet facilement d'incorporer l'incertitude liées aux capteurs ou au modèle. D'autre part, elle est particulièrement adaptée aux opérations de changement de repères. Dans cette approche, la tâche de localisation est décomposée en deux catégories : la localisation incrémentale (ou locale) où l'on suppose que le robot dispose à tout moment d'une estimation de sa position et la localisation globale où le robot est considéré comme « perdu » dans son environnement.

## Localisation incrémentale

Les deux caractéristiques citées plus haut (gestion de l'incertitude et changement de repère) permettent d'effectuer des prédictions en se basant sur des connaissances approximatives de la position et de l'orientation du robot. Une estimation de la position est généralement fournie soit par un système de localisation relative (odométrie) ou par prédiction de l'évolution de la position et de l'orientation en tenant compte des commandes de navigation générées par le système comme dans le système FINALE présenté dans [Kosaka92]. Ces prédictions sont utilisées pour restreindre l'espace de recherche lors de l'étape cruciale de mise en correspondance entre les primitives détectées par les capteurs et celles du modèle en restreignant la recherche aux seules primitives du modèle qui peuvent être détectées dans l'intervalle de position et d'orientation considéré.

## Localisation globale

Dans certaines situations, comme lors de la mise en service du robot ou lors d'un dysfonctionnement du système de localisation relative, le robot ne dispose d'aucune estimation de la position et de l'orientation. La mise en correspondance devient alors plus complexe. En effet le nombre de combinaisons à envisager explose et le risque de faux appariements est élevé à cause des symétries et des ressemblances entre différentes parties du modèle global. La complexité est donc liée à la taille du modèle considéré et au degré de symétrie de ses composants. La solution est l'accumulation d'un grand nombre de données en multipliant les mesures capteurs jusqu'à lever toute ambiguïté. Ceci peut, bien entendu, être coûteux en temps de calcul. Une méthode est présentée dans [Jensfelt01]. Elle utilise le suivi de position (même principe que la localisation incrémentale) basée sur des hypothèses multiples. Cette approche est une adaptation d'une technique employée en suivi de cible par radar au domaine de la robotique mobile.

## Localisation grâce aux grilles d'occupation

L'espace 2D de déplacement du robot est décomposé en cellules qui peuvent être de taille et de forme fixes ou variables. A chaque cellule est associé un état correspondant à la présence ou non du robot dans cette cellule. La liste de ces états doit ensuite être maintenue à jour. La précision de la localisation dépend donc de la résolution du découpage de l'espace global. Mais une trop grande résolution risque d'induire des routines de mise à jour trop coûteuses en temps de calcul.

La localisation grâce à des grilles d'occupation, bien qu'étant discutable en terme de précision, présente l'avantage d'être très peu dépendante de la nature du capteur utilisé [Hager90]. Plusieurs techniques de décompositions existent. Elles peuvent être regroupées en deux catégories, les méthodes exactes et les méthodes approchées ([Hoppenot97]). Le calcul de localisation quant à lui se fait grâce à un regroupement d'informations capteurs. Les cellules ayant une forte probabilité d'être occupées sont retenues.

Plusieurs méthodes de mise à jour de l'état des cellules existent. Dans [Borenstein91] une méthode simple permettant de réduire le temps de calcul de la mise à jour est présentée. Elle consiste à ne traiter qu'un sous ensemble de cellules déterminé par l'axe acoustique du sonar. Dans [Mouaddib00] l'environnement est décomposé en cellules de forme rectangulaire. Un capteur goniométrique permet d'acquérir un certain nombre de données représentant les angles sous lesquels sont vus les amers de l'environnement. Chaque cellule est alors testée grâce à une contrainte sur l'orientation du robot. La cellule retenue est elle même décomposée en cellules plus petite et l'opération est répétée jusqu'à atteindre la précision désirée.

## Localisation basée sur le modèle grâce à la vision

Après avoir survolé les aspects de la localisation en robotique mobile puis de la localisation en environnement intérieur, nous nous focalisons à présent sur le problème de la localisation basée sur la vision. Autrement dit, nous nous intéressons au cas où le capteur utilisé est une caméra. Tous les aspects de la problématique de localisation énoncés plus hauts restent valables. Il est difficile d'établir un état de l'art complet des travaux de localisation basée sur la vision tant la liste de ceux-ci est longue. Nous nous contenterons donc de citer les principaux groupes auxquels peuvent être associés la grande majorité des travaux publiés. A partir des premiers robots mobiles dotés de vision par ordinateur, très vite, la nécessité de fournir aux systèmes de vision une connaissance a priori de ce que le robot est sensé voir s'est imposé. Cette connaissance pouvait se présenter sous forme de modèles métriques (primitives géométriques ou grilles d'occupation), de modèle topologique ou encore sous forme de base d'images. DeSouza ([DeSouza02]) regroupent les approches existantes en trois catégories (qui sont) :

- **Localisation (ou navigation) basée sur le modèle.** Le système utilise un modèle prédéfini conçu par l'humain.
- **Reconstruction de modèle.** Le système construit grâce à ses propres capteurs un modèle de son environnement puis l'utilise pour se localiser.
- **Navigation sans modèle.** Le système n'utilise aucune représentation explicite de l'environnement mais se base sur le tracking d'objets reconnus sur les prises de vue.

Dans les sections suivantes, nous nous intéresserons à l'approche de localisation basée sur un modèle à primitives géométriques puisque les travaux de cette thèse s'inscrivent dans ce registre.

L'idée centrale est de fournir au système de navigation un ensemble de primitives potentiellement visibles et détectables par des techniques de vision par ordinateur durant l'opération de navigation. La tâche du système consiste donc à rechercher puis identifier ces primitives dans l'image. Si un nombre minimal de primitives dans la scène observée est identifiée le système peut calculer la position et l'orientation à partir de laquelle sont vues ces primitives. Le processus classique de localisation grâce à la vision se présente comme une séquence de quatre étapes qui sont

- **Acquisition de l'information capteur.** Le robot saisit une image de son environnement à partir de sa position courante.
- **Détection de primitives (ou amers).** Cette étape correspond généralement aux opérations de traitement d'image telles que l'extraction de contours et la segmentation.
- **Mise en correspondance des observations avec le modèle.** Trouver pour chaque primitive de l'image la primitive correspondante dans le modèle si elle existe.
- **Calcul de la position et de l'orientation de la caméra (donc du robot).** Se fait grâce aux correspondances établies dans l'étape précédente.

Deux types de connaissances *a priori* sont nécessaires pour réaliser les quatre étapes. Il s'agit du modèle de l'environnement et du modèle du capteur. Chacune des trois dernières étapes de localisation et chacune des connaissances *a priori* représente en soit un vaste domaine de recherche sur lesquelles il existent un grand nombre de travaux publiés. Dans ce qui suit nous présentons brièvement le rôle de chaque étape.

### Connaissances *a priori*

## Modélisation de l'environnement

Nous avons abordé dans la section 1.2.2 les modèles de l'environnement à base de primitives géométriques. Nous avons vu que ce modèle peut être soit construit par l'humain et fourni au système soit construit grâce aux capteurs extéroceptifs du robot. Dans les deux cas le modèle global est désigné de façon générique par un ensemble  $\mathbf{Mg} = \{M_1, M_2, \dots, M_m\}$  de primitives. Dans les applications orientées vision, les segments, surfaces ou entités volumiques sont généralement les plus utilisés car d'une part ils sont bien adaptés à la représentation d'un environnement d'intérieur qualifié généralement de semi structuré et d'autre part ils sont détectables par traitement d'image (segmentation en contours, régions, textures).

## Modélisation du capteur

Les données observées sont représentées dans un espace à deux dimensions alors que l'environnement est modélisé dans un repère à trois dimensions. Pour permettre la mise en correspondance entre une primitive donnée de l'image et une primitive du modèle ou le calcul de la position et de l'orientation de la caméra il faut modéliser mathématiquement le processus de formation de l'image. Autrement dit, choisir les équations mathématiques de projection d'un point de l'espace sur le plan de l'image et identifier leurs paramètres ([Faugeras93]).

Plusieurs modèles de caméra existent. Ils diffèrent par plusieurs aspects. Le premier de ces aspects concerne le degré de prise en compte de l'effet de perspective. On distingue en effet, les modèles à perspective pleine et les modèles affines (perspective faible et projection orthographique) [Horaud93]. Un autre aspect est la prise en compte ou non des distorsions. Le choix d'un modèle doit tenir compte de la qualité de la caméra et des dimensions des primitives à observer par rapport à leur distance au centre optique.

Une fois un modèle choisi, il reste à identifier expérimentalement les valeurs de ses paramètres pour représenter le plus fidèlement possible le canal physique d'acquisition. C'est l'opération de calibrage. Les nombreuses méthodes de calibrage peuvent être regroupées en deux grandes familles : les méthodes de calibrage grâce à des objets de référence et les méthodes de calibrage automatiques (ou auto-calibrage) ([Zhang00]). La précision de l'identification du modèle détermine celle du calcul de la position et de l'orientation.

## Etapas de la localisation

### Extraction des primitives de la scène observée

Le but de cette étape est de distinguer parmi la grande quantité d'informations que véhicule une image, celles qui peuvent correspondre à des primitives du modèle. Un pré-traitement de l'image vise à améliorer le rapport signal/bruit (filtrage, rehaussement de contraste). Ensuite, les contours sont extraits puis segmentés en primitives géométriques choisies. Ce traitement permet de définir chaque primitive par un ensemble d'attributs (dimensions, position, texture ou couleur du voisinage, ...). La scène observée est ainsi modélisée par un ensemble  $\mathbf{C} = \{C_1, C_2, \dots, C_n\}$ . Le temps de calcul est un problème crucial dans cette étape. Le choix notamment des algorithmes de chaînage et d'approximation polygonale des contours est très déterminant.

### Mise en correspondance des primitives observées avec celles du modèle



C'est sans nul doute l'opération la plus difficile à exécuter. En effet la mise en correspondance est le problème central dans tout système de vision. Il l'est d'autant plus si ce système est basé sur la vision monoculaire. Ceci explique le nombre impressionnant de travaux publiés dans le domaine. Une grande partie de ces travaux sont du domaine de la reconnaissance d'objets. Ils peuvent être adaptés au domaine de la localisation basée sur un modèle en considérant l'environnement modélisé du robot comme un cas extrême d'objet à reconnaître.

La quasi totalité des approches peuvent être regroupées en deux grandes familles : les méthodes cherchant dans l'espace des correspondances, et celles cherchant dans l'espace des transformations. Le but est de retrouver un nombre suffisant d'appariements cohérents entre les primitives de la scène et celles du modèle. Ce nombre est dépendant de la complexité de la scène et du modèle. Il définit le nombre minimum de données requises pour lever les ambiguïtés de reconnaissance.

Les aspects à prendre en compte dans cette étape sont multiples. Premièrement, les données de l'image sont 2D alors que celles du modèle sont généralement en 3D. La comparaison entre les deux ensembles de primitives ne peut être effectuée que si l'on dispose d'une hypothèse de transformation rigide entre le repère du modèle et celui de l'image. Ceci permet de projeter les primitives du modèle dans la scène. Dans ce cas, il reste à quantifier la ressemblance entre deux ensembles de primitives même définis dans le même espace. Un autre aspect est que plusieurs primitives de la scène sont en fait dues au bruit et aux défauts de segmentation. Ensuite certaines primitives du modèle que l'on s'attend à retrouver dans la scène ne sont pas présentes à cause des occultations et des défauts de détection. Et enfin, et c'est le point le plus important, il faut que la stratégie de recherche soit suffisamment efficace pour contrecarrer l'explosion du nombre de possibilité de correspondances et de transformations.

## **Calcul de l'orientation et de la position du robot**

Après avoir établi une correspondance entre les primitives de la scène et celles du modèle, il reste à calculer la position et l'orientation de la caméra par rapport au repère du modèle. Nous nous intéressons ici aux méthodes de calcul de la pose de la caméra grâce à la vision monoculaire. Autrement dit, et contrairement au cas d'un capteur stéréoscopique, la notion de profondeur des primitives observées est absente.

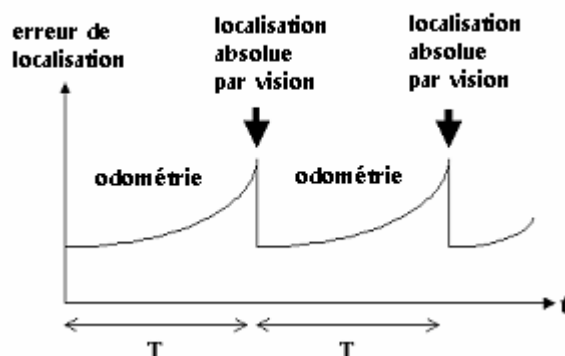
Les méthodes existantes appartiennent à deux catégories : les méthodes analytiques et les méthodes numériques basées sur l'optimisation. La nature bruitée des mesures (surtout les données de l'image) font que les méthodes numériques sont mieux adaptées. Les équations de projection sont fortement non linéaires surtout si l'on utilise un modèle de caméra à perspective pleine. Il faut alors gérer la sensibilité des méthodes de calcul numérique aux conditions initiales. Ces méthodes donnent de bons résultats lorsque les systèmes d'équations sont surdéterminés. Le nombre d'appariements disponibles est donc également à prendre en compte.

## Conclusions et directions de recherche

Dans les sections précédentes, nous avons présenté brièvement la problématique de la localisation en robotique mobile ainsi que l'éventail d'approches et de techniques utilisées pour l'effectuer. Nous nous sommes ensuite focalisé sur la robotique dans un environnement intérieur puis plus précisément sur la localisation orientée vision.

Premièrement il ressort de cette étude l'aspect de complémentarité entre les deux grandes approches de localisation que sont la localisation relative et la localisation absolue. En effet, l'odométrie, système de localisation relative par excellence, dépend de l'origine et possède une erreur non bornée dans le temps mais elle est très simple à mettre en œuvre et permet un très haut niveau d'échantillonnage. A l'inverse, les techniques de localisation absolue sont à erreur bornée mais sont beaucoup plus lourde à implanter et pour la plupart ne permettent que des niveaux d'échantillonnage limités. Pour exploiter les avantages des deux approches et avoir un système de localisation performant, il semble judicieux de combiner deux techniques, une de chaque catégorie.

L'objectif fixé est donc le développement d'un système de localisation incrémentale en ligne. L'odométrie est utilisée pour maintenir à jour sa connaissance de la position et de l'orientation avec un haut niveau d'échantillonnage. Le système de vision monoculaire vient périodiquement recalibrer les mesures odométriques pour maintenir l'erreur de localisation à un niveau inférieur à un seuil prédéfini (figure 1.8).



*Figure 1.8 : approche de localisation combinant l'odométrie et la vision*

Nous avons vu que la localisation grâce à la vision dépendait de la qualité de l'estimation initiale de la position et de l'orientation. Ici, cette estimation est fournie par l'odométrie. Sa qualité dépend donc de la fréquence avec laquelle sont effectués les recalages. En effet, plus la période  $T$  est longue plus la divergence du système odométrique est grande. D'un autre côté, la localisation basée sur le modèle a un coût calculatoire important du fait de la succession de plusieurs étapes de calcul. Ceci est d'autant plus vrai pour la localisation grâce à la vision car chaque étape (acquisition, pré-traitement, extraction de primitives de la scène, mise en correspondance et calcul de la position et de l'orientation) nécessite dans ce cas le traitement d'une grande quantité de données. La fréquence ne doit donc pas être trop élevée pour que le système fonctionne en ligne.

Un autre critère à prendre en compte lors d'un fonctionnement hors ligne est la disponibilité ou non d'un nombre suffisant de données pour effectuer un recalage fiable. En effet, il peut arriver que pour une acquisition donnée, le nombre de primitives du modèle potentiellement visibles dans la scène soit trop faible ou que l'extraction des primitives de l'image ne donne pas un résultat satisfaisant (saturation en nombre de segments détectés ou pas assez de segments fiables détectés en raison de mauvaises conditions d'éclairage par exemple). Dans

ce cas il est préférable que le système reporte le calcul de recalage à une acquisition ultérieure. La durée  $T$  s'en trouve donc prolongée et la qualité de l'initialisation est amoindrie.

Lors du développement du système de localisation grâce à la vision monoculaire, nous avons visé comme objectif d'optimiser les performances de chacune des étapes de localisation en terme de sensibilité aux conditions initiales, de temps de calcul, de robustesse en présence de données bruitées afin que la chaîne complète soit performante. En terme de précision, la plupart des systèmes de vision, (dans un domaine du même ordre de grandeur que celui d'un environnement d'intérieur : quelques mètres voire quelques dizaines de mètres), permettent d'atteindre une précision d'environ  $3^\circ$  pour l'orientation et environ 10 cm pour la position ([Borenstein98]). Cette précision est suffisante pour des missions de navigation autonomes et peut servir de référence pour l'évaluation de notre travail.

Les chapitres suivants concernent les choix effectués et les contributions dans chacune des étapes du processus de localisation grâce à la vision monoculaire énoncés ci-dessus.



# Outils pour la localisation par vision monoculaire

Pour que le processus de localisation puisse être mené à bien certaines connaissances doivent être établies hors-ligne. Il s'agit du modèle de l'environnement et du modèle du capteur. Un outil de traitement de l'image acquise en vue d'extraire des primitives segmentées de l'information est également indispensable. Ce chapitre regroupe les choix arrêtés et les travaux réalisés dans chacun de ces domaines.

La première partie concerne la modélisation de la caméra et la mise en œuvre du calibrage de la caméra par la méthode Zhang ([Zhang00]). La présentation des différents modèles de caméra est nécessaire pour la compréhension des chapitres suivants. Le choix de la méthode de Zhang est justifié par rapport au contexte de la robotique mobile. La seconde partie est consacrée à la construction du modèle de l'environnement. Le choix retenu est un modèle filaire complété d'un découpage en régions d'invariance visuelle pour la gestion des occultations. En fin de chapitre sont présentés les algorithmes utilisés pour le pré-traitement de l'image, l'extraction de contours et leur segmentation.

# Modélisation et calibrage du capteur

## Etat de l'art

Nous entendons par « modèle de caméra », l'ensemble de lois géométriques définissant la façon dont se projette, lors du processus de saisie d'une image, un point de l'espace à trois dimensions sur un plan à deux dimensions. Un modèle est caractérisé par un certain nombre de paramètres permettant de calculer, à partir des coordonnées cartésiennes d'un point de l'espace, les coordonnées en pixels sur l'image. Selon la précision désirée, plusieurs modèles ont été proposés prenant en compte, plus ou moins fidèlement, la façon dont se déplace le faisceau de lumière pour constituer l'image ([Faugeras93], [Horaud93]).

Une fois un type de modèle choisi, il faut identifier ses paramètres. L'estimation de la valeur de ces paramètres pour une caméra est réalisée par calibrage. C'est une étape nécessaire pour toute application orientée vision. De nombreuses techniques ont été développées. Elles peuvent être classées en deux catégories :

- **Calibrage basé sur des objets 3D de référence.**

Cette technique utilise l'observation d'objets 3D de coordonnées connues. Les objets de calibrage (mire) sont généralement des points répartis sur des plans orthogonaux ou sur un plan translaté dans la direction de sa normale. Le calcul peut alors être effectué de façon relativement simple ([Puget90], [Horaud93], [Zhang00], [Batista99]).

- **Calibrage automatique (ou auto-calibrage).**

Le mouvement connu de la caméra filmant une scène statique est utilisé pour poser des contraintes sur les paramètres intrinsèques prenant en compte la rigidité des objets filmés en utilisant uniquement les informations de l'image ([Du96], [Hartley94]). Bien que très flexible cette deuxième approche n'est pas encore assez développée pour obtenir des résultats satisfaisants.

Dans ce qui suit, les modèles de projections les plus connus sont présentés, puis quelques techniques de calibrage basées sur des objets de référence sont explicitées. Il s'agit notamment de la méthode de Faugéras-Toscani et d'une méthode plus flexible et plus adaptée aux applications où le volume 3D contenant les objets de référence doit être important (le cas de la robotique mobile en milieu intérieur en est un exemple).

## Modèles de caméra

### Modèle de projection perspective

La projection d'un point de l'espace sur une image dépend, d'une part, de sa localisation par rapport à la caméra et d'autre part des caractéristiques physiques du canal d'acquisition (objectif de la caméra).

Pour formaliser ces deux notions, trois systèmes de coordonnées sont définis (figure 2.1) :

- un repère 3D appelé repère monde (ou repère objet)  $O(x,y,z)$  dans lequel sont exprimés les points 3D des objets constituant la scène filmée,
- un repère  $O_c(x_c,y_c,z_c)$  3D lié à la caméra de telle façon que l'origine soit le centre optique et l'axe  $z_c$  confondu avec le rayon optique.

- un repère 2D lié au plan image, parallèle au plan  $u-v$  et situé à une distance  $f$  (distance focale) de  $O_c$ .

Adoptant la représentation en coordonnées homogènes, un point arbitraire  $P[x,y,z,1]^T$  exprimé dans le repère objet se projette sur le plan image en un pixel  $I[u,v,1]^t$  grâce à une transformation représentée par la matrice  $M$  telle que

$$\lambda \cdot I = MP \quad (2.1.1)$$

où  $\lambda$  est un facteur d'échelle scalaire.

La matrice de transformation globale  $M$  est composée d'une matrice homogène  $G$  et d'une matrice de projection perspective  $K$ .  $G$  représente le modèle externe de la caméra et  $K$  le modèle interne.

### Le modèle externe

$G$  est la transformation homogène composée d'une rotation  $R$  et d'une translation  $t$  qui amènent le repère caméra sur le repère objet. Elle permet de transformer le point 3D  $P[x,y,z,1]^T$  du repère objet en un point  $P_c[x_c,y_c,z_c,1]^T$  dans le repère caméra.

Sous forme matricielle :

$$P_c = GP \quad (2.1.2)$$

avec

$$G = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}.$$

Les composantes de  $R$  et  $t$  sont appelés paramètres extrinsèques.

### Le modèle interne

$K$  transforme le point  $P_c[x_c,y_c,z_c,1]^T$  en un pixel  $I[u,v,1]^T$  sur l'image. Cette transformation est elle-même composée d'une projection perspective sur le plan image suivie d'une transformation affine.

Les coordonnées du point de projection perspective par rapport au centre optique sur le plan image s'obtiennent par les équations suivantes

$$\begin{cases} x' = f \cdot \frac{x_c}{z_c} \\ y' = f \cdot \frac{y_c}{z_c} \end{cases} \quad (2.1.3)$$

La transformation affine donnant les coordonnées en pixels est composée par une mise à l'échelle et une translation :

$$\begin{cases} u = -k_u \cdot x' + u_0 \\ v = k_v \cdot y' + v_0 \end{cases} \quad (2.1.4)$$

où  $k_u$  et  $k_v$  représentent les facteurs d'échelle vertical et horizontal respectivement et  $u_0, v_0$  l'origine du repère des coordonnées en pixels.

Sous forme matricielle

$$\lambda \cdot I = KP_c \quad (2.1.5)$$

avec

$$K = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.1.6)$$

où  $\alpha_u = -k_u \cdot f$ ,  $\alpha_v = k_v \cdot f$

Les composantes de  $\mathbf{K}$  sont appelées paramètres intrinsèques de la caméra.

En résumé, le modèle global d'une caméra peut s'écrire comme suit :

$$\lambda \cdot \mathbf{I} = \mathbf{MP} = \mathbf{KGP} \quad (2.1.7)$$

ou, en plus détaillé

$$\lambda \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

En posant  $\mathbf{r}_i = [r_{i1} \ r_{i2} \ r_{i3}]$  pour  $i=1,2,3$ ,  $\mathbf{G}$  s'écrit

$$\mathbf{G} = \begin{bmatrix} \mathbf{r}_1 & t_x \\ \mathbf{r}_2 & t_y \\ \mathbf{r}_3 & t_z \\ 0 & 1 \end{bmatrix}.$$

Une écriture plus compacte de  $\mathbf{M}$  est alors obtenue sous la forme

$$\mathbf{M} = \begin{bmatrix} \alpha_u \mathbf{r}_1 + u_0 \mathbf{r}_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v \mathbf{r}_2 + v_0 \mathbf{r}_3 & \alpha_v t_y + v_0 t_z \\ \mathbf{r}_3 & t_z \end{bmatrix} \quad (2.1.8)$$

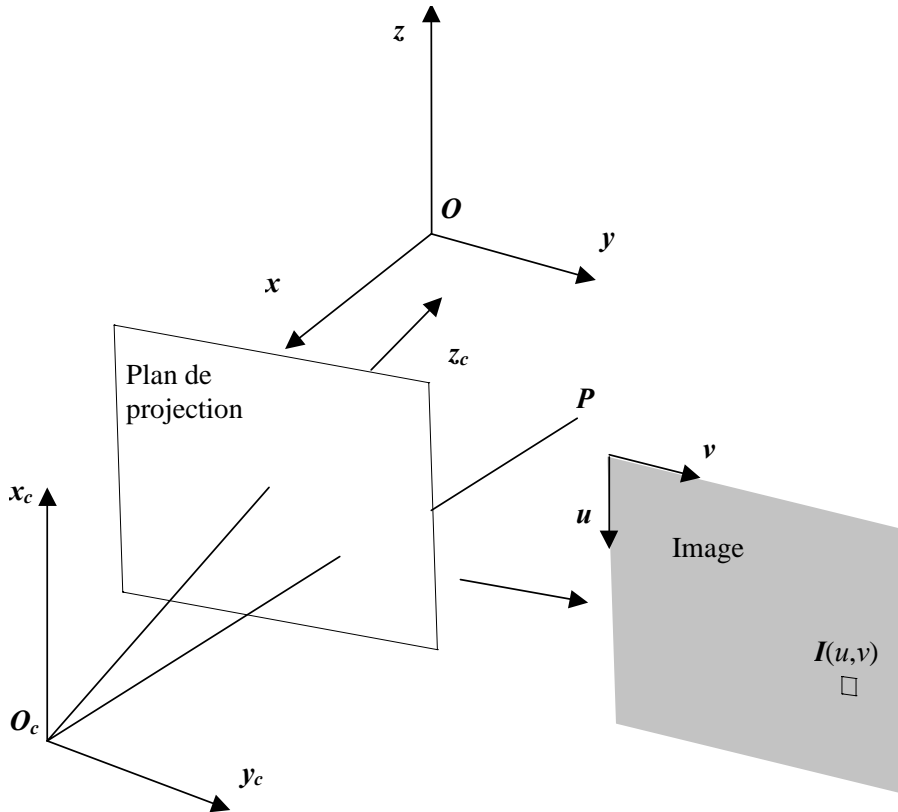
En pratique les vecteurs du repère image en coordonnées pixels ne sont pas tout à fait perpendiculaires. Pour modéliser cet effet, l'angle  $\theta$ , entre les vecteurs  $\mathbf{u}$  et  $\mathbf{v}$ , est introduit dans les équations précédentes. Un nouveau repère de coordonnées image  $\mathbf{u}'\text{-}\mathbf{v}'$  est alors défini et la loi de changement de repère s'écrit comme suit

$$\begin{cases} u' = u \cdot \sin(\theta) + v \cdot \cos(\theta) \\ v' = v \end{cases} \quad (2.1.9)$$

d'où

$$\begin{cases} u_0' = u_0 \cdot \sin(\theta) + v_0 \cdot \cos(\theta) \\ v_0' = v_0 \end{cases} \quad (2.1.10)$$





**Figure 2.1 :** passage du repère de la scène à l'image

La nouvelle matrice de paramètres intrinsèques devient

$$\mathbf{K} = \begin{bmatrix} \alpha_u & \alpha_v \cdot \cot(\theta) & u'_0 & 0 \\ 0 & \frac{\alpha_v}{\sin(\theta)} & v'_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.1.11)$$

Ce qui donne le nouveau modèle global de la caméra suivant :

$$\mathbf{M} = \begin{bmatrix} \alpha_u \mathbf{r}_1 + \alpha_u \cot(\theta) \mathbf{r}_2 + u'_0 \mathbf{r}_3 & \alpha_u t_x + \alpha_u \cot(\theta) t_y + u'_0 t_z \\ \frac{\alpha_v}{\sin(\theta)} \mathbf{r}_2 + v'_0 \mathbf{r}_3 & \frac{\alpha_v}{\sin(\theta)} t_y + v'_0 t_z \\ \mathbf{r}_3 & t_z \end{bmatrix} \quad (2.1.12)$$

### Modèle affine

Le modèle de projection perspective ci-dessus est un modèle général mais complexe a cause du grand nombre de degrés de liberté (10) mais surtout à cause de sa non linéarité.

Pour plus de simplicité, supposons que le modèle interne soit représenté par une matrice

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

ce qui n'enlève rien à la généralité.

Les coordonnées pixel d'un point arbitraire  $\mathbf{P}$  s'expriment alors

$$\begin{cases} u = \frac{\mathbf{P} \cdot \mathbf{r}_1 + t_x}{\mathbf{P} \cdot \mathbf{r}_3 + t_z} \\ v = \frac{\mathbf{P} \cdot \mathbf{r}_2 + t_y}{\mathbf{P} \cdot \mathbf{r}_3 + t_z} \end{cases} \quad (2.1.13)$$

Le dénominateur commun du terme de droite des deux équations dépend non seulement de la translation entre les deux repères caméra et objet dans la direction z mais aussi de la position de chaque point symbolisé ici par  $\mathbf{P} \cdot \mathbf{r}_3$ . Une façon d'éviter ce type de non-linéarité est d'établir un modèle global, connu sous le nom de modèle affine, de la forme suivante

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1.14)$$

Deux modèles peuvent correspondre à cette forme de  $\mathbf{M}$ , il s'agit du modèle de projection à perspective faible et du modèle de projection para-perspective.

### Modèle de projection à perspective faible

Mettant en facteur  $t_z$  dans les équations de projection perspective précédentes on obtient

$$\begin{cases} u = \frac{\mathbf{P} \cdot \frac{\mathbf{r}_1}{t_z} + \frac{t_x}{t_z}}{1 + \varepsilon} \\ v = \frac{\mathbf{P} \cdot \frac{\mathbf{r}_2}{t_z} + \frac{t_y}{t_z}}{1 + \varepsilon} \end{cases} \quad (2.1.15)$$

avec  $\varepsilon = \mathbf{P} \cdot \frac{\mathbf{r}_3}{t_z}$

Une approximation d'ordre zéro remplaçant  $\frac{1}{1 + \varepsilon}$  par 1 permet d'obtenir le modèle à perspective faible représenté par les équations

$$\begin{cases} u = \frac{\mathbf{P} \cdot \mathbf{r}_1 + t_x}{t_z} \\ v = \frac{\mathbf{P} \cdot \mathbf{r}_2 + t_y}{t_z} \end{cases} \quad (2.1.16)$$

L'interprétation géométrique de cette approximation est représentée par la figure 2.2.

Les points de l'objet 3D sont d'abord projetés orthogonalement sur un plan parallèle au plan image, puis un facteur d'échelle de  $\frac{1}{t_z}$  est appliqué sur les points ainsi obtenus. Ceci revient à considérer que tous les points de la scène filmée ont en commun la même composante z dans le repère caméra ou encore que les différences entre ces composantes sont négligeables devant la distance  $t_z$  au centre optique ce qui a pour effet d'atténuer les effets de perspective.

### Modèle de projection para-perspective

Ce modèle est obtenu grâce à une approximation d'ordre 1 du modèle projectif initial, c'est à dire remplacer  $\frac{1}{1 + \varepsilon}$  par  $1 - \varepsilon$ .

Les équations de projection deviennent

$$\begin{cases} u = \frac{P \cdot (\mathbf{r}_1 - \frac{t_x}{t_z} \mathbf{r}_3) + t_x}{t_z} \\ v = \frac{P \cdot (\mathbf{r}_2 - \frac{t_y}{t_z} \mathbf{r}_3) + t_y}{t_z} \end{cases} \quad (2.1.17)$$

L'interprétation géométrique est sur la figure 2.2. Dans ce cas les points sont projetés sur un plan parallèle au plan image, non de façon orthogonale, mais en parallèle à la droite  $O_c O_m$  où  $O_m$  est l'origine du repère objet. Un facteur d'échelle de  $\frac{1}{t_z}$  est ensuite appliqué sur les points ainsi obtenus. Là aussi, le modèle est d'autant plus réaliste que les dimensions de l'objet filmé sont négligeables devant leur distance au centre optique.

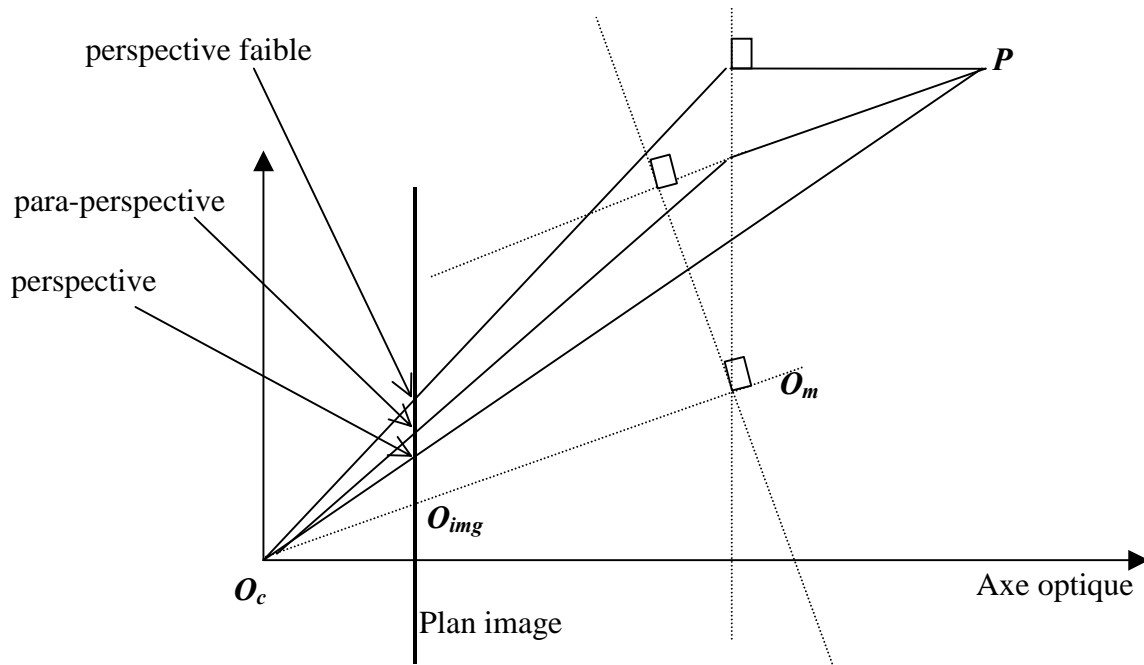


Figure 2.2 : les trois différents modèles de projection

### Modèle avec prise en compte des distorsions

En réalité le faisceau lumineux qui projette un point spatial sur le plan image n'est pas tout à fait une droite. Une correction radiale doit alors être apportée aux coordonnées pixels obtenues. Les équations de projection deviennent alors

$$\begin{cases} u = \alpha_u \cdot x + u_0 + \Delta u \\ v = \alpha_v \cdot y + v_0 + \Delta v \end{cases} \quad (2.1.18)$$

avec

$$\begin{cases} \Delta u = k((u - u_0)^2 + (v - v_0)^2)(u - u_0) \\ \Delta v = k((u - u_0)^2 + (v - v_0)^2)(v - v_0) \end{cases} \quad (2.1.19)$$

Le modèle interne comprend donc un paramètre supplémentaire  $k$ .

## Calibrage de caméra

Calibrer une caméra consiste à identifier les valeurs pratiques de ses paramètres constituant le modèle théorique. Ils s'agit plus spécialement des paramètres intrinsèques car ils constituent la géométrie du capteur lui-même (objectif, barrette CCD). Un changement de mise au point ou de focale implique bien entendu que le calibrage doit être refait.

### Mise en œuvre du calibrage

A partir de l'équation du modèle global on obtient :

$$\begin{cases} u = \frac{m_{11}x + m_{12}y + m_{13}z + m_{14}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \\ v = \frac{m_{21}x + m_{22}y + m_{23}z + m_{24}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \end{cases} \quad (2.1.20)$$

Ces équations sont linéaires par rapport aux coefficients  $m_{ij}$ . Il faut donc au moins 12 équations de ce genre pour déterminer les 12 coefficients de  $\mathbf{M}$ . Autrement dit, il faut connaître au moins six points dans le repère objet ainsi que leurs projections respectives sur l'image. Pour  $n$  points connus on obtient donc  $2n$  équations de la forme

$$\begin{aligned} m_{11}x_i + m_{12}y_i + m_{13}z_i + m_{14} - u_i m_{31}x_i - u_i m_{32}y_i - u_i m_{33}z_i &= u_i m_{34} \\ m_{21}x_i + m_{22}y_i + m_{23}z_i + m_{24} - v_i m_{31}x_i - v_i m_{32}y_i - v_i m_{33}z_i &= v_i m_{34} \end{aligned} \quad (2.1.22)$$

Ce système d'équation s'écrit sous forme matricielle

$$\begin{bmatrix} x_i & y_i & z_i & 1 & 0 & 0 & 0 & 0 & -u_i x_i & -u_i y_i & -u_i z_i \\ 0 & 0 & 0 & 0 & x_i & y_i & z_i & 1 & -v_i x_i & -v_i y_i & -v_i z_i \\ \vdots & & & & & & & & & & & \vdots \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} \vdots \\ u_i m_{34} \\ v_i m_{34} \\ \vdots \end{bmatrix}$$

que l'on peut écrire de façon plus compacte

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \quad (2.1.21)$$

où  $\mathbf{x}$  est l'inconnue recherchée.

La résolution de cette équation nous donne les valeurs des  $m_{ij}$ . Il reste alors à calculer les paramètres intrinsèques et extrinsèques en fonction de ces valeurs. Cela peut être fait en identifiant le produit  $\mathbf{KG}$  à  $\mathbf{M}$ . On obtient alors en posant  $\mathbf{m}_i = [m_{i1} \ m_{i2} \ m_{i3}]$  pour  $i=1,2,3$

$$\left\{ \begin{array}{l} \mathbf{r}_3 = \mathbf{m}_3 \\ u_0 = \mathbf{m}_1 \cdot \mathbf{m}_3 \\ v_0 = \mathbf{m}_2 \cdot \mathbf{m}_3 \\ \alpha_u = -\|\mathbf{m}_1 \times \mathbf{m}_3\| \\ \alpha_v = \|\mathbf{m}_2 \times \mathbf{m}_3\| \\ \mathbf{r}_1 = \frac{1}{\alpha_u} (\mathbf{m}_1 - u_0 \cdot \mathbf{m}_3) \\ \mathbf{r}_2 = \frac{1}{\alpha_u} (\mathbf{m}_2 - u_0 \cdot \mathbf{m}_3) \\ t_x = \frac{1}{\alpha_u} (m_{14} - u_0 \cdot m_{34}) \\ t_y = \frac{1}{\alpha_u} (m_{24} - u_0 \cdot m_{34}) \\ t_z = m_{34} \end{array} \right. \quad (2.1.23)$$

Le problème revient donc en pratique déterminer  $\mathbf{M}$ .

### Calcul des coefficients de $\mathbf{M}$

#### Utilisation de la contrainte $m_{34}=1$

Une façon de procéder consiste à imposer  $m_{34}=1$  ([Heraud93]). Ceci permet de calculer les autres coefficients  $m_{ij}$  à un facteur d'échelle près en résolvant l'équation linéaire ainsi obtenue. La valeur recherchée de  $\mathbf{x}$  est la solution du système linéaire dans le sens des moindres carrés :

$$\mathbf{x} = (\mathbf{A}^t \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^t \cdot \mathbf{u} \quad (2.1.24)$$

#### Méthode de Faugeras-Toscani

En remarquant que  $\|\mathbf{m}_3\|=1$  ([Faugeras86]), l'équation précédente peut se décomposer comme suit

$$\mathbf{B}\mathbf{x}_1 + \mathbf{C}\mathbf{x}_2 = 0 \quad (2.1.25)$$

avec

$$\mathbf{B} = \begin{bmatrix} & & & & \vdots & & & & & \\ x_i & y_i & z_i & 1 & 0 & 0 & 0 & 0 & -u_i & \\ 0 & 0 & 0 & 0 & x_i & y_i & z_i & 1 & -v_i & \\ & & & & \vdots & & & & & \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} \vdots \\ -u_i x_i & -u_i y_i & -u_i x_i \\ -v_i x_i & -v_i y_i & -v_i x_i \\ \vdots \end{bmatrix}$$

$$\mathbf{x}_1 = [m_1 \quad m_{14} \quad m_2 \quad m_{24} \quad m_{34}]^T$$

$$\mathbf{x}_2 = \mathbf{m}_3^T$$

Soit  $Q = \|\mathbf{B}\mathbf{x}_1 + \mathbf{C}\mathbf{x}_2\|^2$  le critère quadratique à minimiser avec la contrainte  $\|\mathbf{x}_2\|=1$ , ce qui se traduit par  $Q = \|\mathbf{B}\mathbf{x}_1 + \mathbf{C}\mathbf{x}_2\|^2 + \lambda(1 - \|\mathbf{x}_2\|^2)$ . En imposant que les dérivées partielles de  $Q$  par rapport à  $\mathbf{x}_1$  et  $\mathbf{x}_2$  soient nulles et en substituant dans l'expression de  $Q$  on trouve  $Q = \mathbf{x}_2^t \mathbf{D} \mathbf{x}_2$  avec  $\mathbf{D} = \mathbf{C}^t \mathbf{C} - \mathbf{C}^t \mathbf{B} (\mathbf{B}^t \mathbf{B})^{-1} \mathbf{B}^t \mathbf{C}$ .  $\mathbf{D}$  étant symétrique et définie positive,  $\mathbf{x}_2$  n'est autre que le vecteur propre de  $\mathbf{D}$  associée à la valeur propre  $\lambda$ . Il suffit donc de calculer la plus petite valeur propre de  $\mathbf{D}$  puis le vecteur propre associé pour obtenir  $\mathbf{x}_2$ .  $\mathbf{x}_1$  est ensuite obtenu grâce à l'équation suivante  $\mathbf{x}_1 = -\mathbf{B} (\mathbf{B}^t \mathbf{B})^{-1} \mathbf{B}^t \mathbf{C} \mathbf{x}_2$  qui découle de l'annulation des dérivées partielles.

Il est à noter que l'ensemble des  $n$  points considérés dans le repère objet ne doivent pas être coplanaires car la méthode présente dans ce cas une singularité.

Dans le cas d'un modèle avec prise en compte des distorsions, les équations précédentes ne conduisent plus à un système d'équations linéaires par rapport aux coefficients  $m_{ij}$ .

L'utilisation d'une méthode d'optimisation non linéaire s'impose alors. L'estimation initiale des valeurs des paramètres, nécessaire à ce type d'optimisation, peut être trouvée en adoptant, dans un premier temps, une approche linéaire.

Les équations permettant l'obtention des paramètres intrinsèques et extrinsèques à partir des coefficients de  $\mathbf{M}$  sont non linéaires. Les erreurs commises sur les valeurs des coefficient  $m_{ij}$  peuvent alors se propager de façon très défavorables sur les valeurs des paramètres (en particuliers  $u_0$  et  $v_0$ ). Les mesures pratiques des coordonnées des points dans le repère objet ainsi que l'extraction de leurs projections respectives sur l'image grâce à la segmentation de celle-ci doivent être aussi précise que possible. L'opération de calibrage nécessite donc un matériel assez pointu. Il faut notamment disposer d'une mire avec un certain nombre de points et d'un moyen de déplacer ces points parfaitement orthogonalement au plan x-y du repère objet pour disposer de points non coplanaires.

### Une méthode de calibrage plus flexible ([Zhang00])

La méthode présentée ici requiert deux ou plus images d'un ensemble de points coplanaires. La connaissance du mouvement entre les plans de chaque image n'est pas nécessaire. Les images peuvent être réalisées simplement grâce à une impression laser collée sur une surface raisonnablement plane.

Soit  $\mathbf{M}=[x,y,z,1]^T$  un point 3D dans le repère objet et  $\mathbf{m}=[u,v,1]^T$  sa projection sur l'image.

L'équation de projection s'écrit

$$s.\mathbf{m} = \mathbf{A}[\mathbf{R} \ \mathbf{T}]\mathbf{M} \quad (2.1.26)$$

avec  $\mathbf{A} = \begin{bmatrix} \alpha & \beta & u_0 \\ 0 & \gamma & v_0 \\ 0 & 0 & 1 \end{bmatrix}$  où  $\alpha$  et  $\beta$  désignent les paramètres internes précédemment définis

complétés par le paramètre de codage de l'angle entre les axes de l'image  $\gamma$ .

En supposant que les points du plan image ont en commun la même coordonnée  $z=0$  et par abus de notation on désigne les points  $M$  par  $[x,y,1]^T$ .  $M$  et  $m$  sont alors liés par la relation suivante :

$$s.m = HM \quad (2.1.27)$$

où  $H$  est une homographie définie par  $H = A[r_1 \ r_2 \ T]$ .

A partir d'une image du plan modèle une homographie  $H=[h_1 \ h_2 \ h_3]$  peut être estimée.

L'orthonormalité de la matrice de rotation produit les contraintes suivantes :

$$h_1^t A^{-t} A^{-1} h_2 = 0 \quad (2.1.28)$$

$$h_1^t A^{-t} A^{-1} h_1 = h_2^t A^{-t} A^{-1} h_2 \quad (2.1.29)$$

On pose  $B=A^t A^{-1}$ .

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - y_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - y_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \quad (2.1.30)$$

$B$  est symétrique définie positive.

Soit le vecteur  $b=[B_{11} \ B_{12} \ B_{13} \ B_{11} \ B_{12} \ B_{13}]$ . Soit  $h_i=[h_{i1} \ h_{i1} \ h_{i1}]^t$  la  $i^{\text{ème}}$  colonne de  $H$ . On a alors

$$h_i^t B h_i = v_{ij}^t b \quad (2.1.31)$$

avec  $v_{ij}^t=[h_{i1}h_{j1}, h_{i1}h_{j1}, h_{i1}h_{j1}, h_{i1}h_{j1}, h_{i1}h_{j1} + h_{i1}h_{j1}, h_{i1}h_{j1} + h_{i1}h_{j1}, h_{i1}h_{j1}]^t$ . Les contraintes précédentes peuvent alors être écrites sous la forme

$$\begin{bmatrix} v_{i2}^t \\ (v_{i1} - v_{i2})^t \end{bmatrix} b = 0 \quad (2.1.32)$$

Avec  $n$  images disponibles on obtient  $2n$  telles équations. Si  $n \geq 3$  le système obtenu a dans le cas général une solution unique pour  $b$ . Si  $n=2$  on peut poser  $\gamma=0$  et estimer les valeurs d'une partie des composantes de  $b$ .

Une fois  $b$  estimée, les paramètres de la caméra sont calculés grâce aux équations suivantes :

$$\begin{cases} v_0 = (B_{12}B_{13} - B_{11}B_{23}) / (B_{12}B_{13} - B_{12}^2) \\ \lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{12}B_{13})] / B_{11} \\ \alpha = \sqrt{\lambda / B_{11}} \\ \beta = \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\ \gamma = -B_{12}\alpha^2\beta / \lambda \\ u_0 = \gamma v_0 / \alpha - B_{13}\alpha^2 / \lambda \end{cases} \quad (2.1.33)$$

Les paramètres extrinsèques correspondant à chaque image peuvent également être calculés.

## Mise en oeuvre du calibrage de notre capteur

### Choix de la méthode

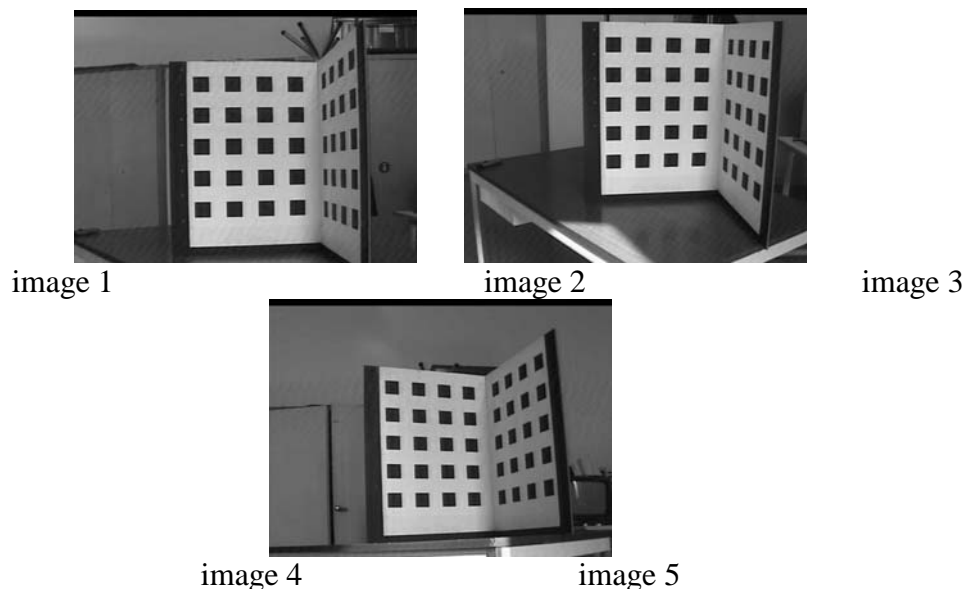
A la lumière de ce bref état de l'art sur les méthodes de calibrage de caméra, il ressort que la méthode de Faugeras-Toscani nécessite que les points de référence utilisés soient non-coplanaires et que les coordonnées soient connues avec une précision de l'ordre du dixième de millimètre ([Heraud93]). La technique employée est généralement un pattern plan déplacé dans une direction orthogonale grâce à un banc de calibrage très précis. Ceci est difficile à réaliser si l'on veut que les paramètres calculés soient valides pour la reconstruction de primitives réparties sur un volume important comme c'est le cas pour les segments d'un environnement de type intérieur. La méthode de Zhang utilise elle un pattern plan photographié à plusieurs position et orientations. Il n'est pas nécessaire de connaître le mouvement du plan entre les différentes prises de vue. Ceci peut être exploité pour utiliser lors du calibrage des points répartis sur un grand volume autour de la caméra. C'est pour cette raison que cette méthode a été retenue pour le calibrage de notre capteur.

La méthode de Zhang a donc été implémentée sous Matlab et le calibrage a été mis en œuvre comme spécifié dans les sections suivantes.

## Protocole expérimental

La méthode de Zhang ([Zhang00]) présentée plus haut a été utilisée pour calibrer une caméra SONY FCB-IX70P avec une résolution d'image de 640X480. Le modèle utilisé est un plan sur lequel a été imprimé un pattern de 4 par 5 carrés noirs. Les points retenus sont les coins de chaque carré. Chaque image du plan fournit ainsi 80 points.

Les 5 images utilisées sont présentées en figure 2.3. Elles montrent le modèle sous différentes orientations et à différentes distances du centre optique. Le plan utilisé est l'un des deux plans de la figure 2.3. Le but est de calculer un ensemble de paramètres valables pour des points répartis dans un espace assez important autour de la caméra comme ce sera le cas pour les primitives observées lors d'une tâche dans un environnement intérieur.



*Figure 2.3 : images utilisées pour le calibrage*

## Résultats

Les paramètres intrinsèques obtenus sont les suivants :



$$\alpha_u = 964.486, \alpha_v = 969.547, u_0 = 274.413, v_0 = 335.482, \gamma = -0.351999,$$

Les matrices de rotation et les vecteurs de translation ( $\mathbf{R}_I, \mathbf{T}_I$ ) décrivant le modèle externe pour chaque image sont les suivantes :

$$\text{Image 1 : } \mathbf{R}_I = \begin{bmatrix} -0.0355 & -0.9993 & -0.0012 \\ 0.7356 & -0.0269 & -0.6768 \\ -0.6764 & 0.0231 & 0.7361 \end{bmatrix} \quad \mathbf{T}_I = [0.242 \quad -0.095 \quad 1.644]^T$$

$$\text{Image 2 : } \mathbf{R}_I = \begin{bmatrix} -0.0216 & -0.9749 & -0.2215 \\ 0.9897 & -0.0103 & -0.1424 \\ -0.1411 & 0.2223 & 0.9646 \end{bmatrix} \quad \mathbf{T}_I = [0.052 \quad -0.129 \quad 1.794]^T$$

$$\text{Image 3 : } \mathbf{R}_I = \begin{bmatrix} -0.0102 & -0.9998 & 0.0155 \\ 0.7663 & -0.0103 & -0.6421 \\ 0.6453 & 0.0053 & 0.7666 \end{bmatrix} \quad \mathbf{T}_I = [0.202 \quad -0.221 \quad 1.451]^T$$

$$\text{Image 4 : } \mathbf{R}_I = \begin{bmatrix} -0.0140 & -0.9997 & 0.0171 \\ 0.9940 & -0.0157 & -0.1034 \\ 0.1037 & 0.0156 & 0.9944 \end{bmatrix} \quad \mathbf{T}_I = [0.201 \quad -0.183 \quad 1.614]^T$$

$$\text{Image 5 : } \mathbf{R}_I = \begin{bmatrix} 0.02183 & -0.9826 & 0.1834 \\ 0.9231 & -0.0446 & -0.3818 \\ 0.3834 & 0.1801 & 0.9056 \end{bmatrix} \quad \mathbf{T}_I = [0.207 \quad -0.205 \quad 1.754]^T$$

On remarque que la valeur de  $\gamma$  qui est le paramètre traduisant l'angle entre les axes de l'image est proche de 0. Ceci signifie que l'angle en question est très proche de  $90^\circ$  comme l'on peut s'y attendre sur ce type de caméra CCD.

### Erreur de reconstruction

Pour évaluer la qualité des paramètres intrinsèques ainsi estimés, les projections de points 3D n'ayant pas participé au calibrage sont calculées en se basant sur les modèles externes obtenus lors du calibrage pour chacune des images. Le degré de superposition des projections calculées sur les pixels de l'image réelle indique la qualité du calibrage. Autrement dit, pour un point 3D  $\mathbf{P}(x,y,z)$ , on calcule sa projection sur l'image  $\mathbf{p}_{mod}(u_{mod}, v_{mod})$  puis on relève le pixel correspondant  $\mathbf{p}_{obs}(u_{obs}, v_{obs})$  observé sur l'image réelle. On calcule ensuite la distance  $\delta = \|(u_{obs} - u_{mod}, v_{obs} - v_{mod})\|$ .

La table 2.1 représente les paramètres statistiques des erreurs de reconstruction en pixels des points 3D. Ces résultats sont les valeurs de  $\delta$  obtenus sur la base de 40 points sélectionnés sur chaque image.

	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5
max.	2,70	4,00	3,30	4,35	3,55
moyenne	0,97	1,56	1,13	1,38	1,09
écart type	0,56	0,63	0,60	0,80	0,73

**Table 2.1:** erreur de reconstruction 2D sur des points n'ayant pas participé au calibrage.

Les résultats montrent que les erreurs de reconstruction sont équivalentes à celles obtenues avec la méthode de Faugéras-Toscani dans [Puget90].



# Modélisation de l'environnement

Le type de modèle adopté pour ce travail est un modèle à base de primitives géométriques. Comme nous l'avons dit précédemment, les avantages d'un tel modèle sont :

- concision dans la représentation de l'information,
- adaptation aux opérations de changement de repères, ce qui permet la prédiction des primitives potentiellement observables à partir d'une estimation de la position.
- possibilité de modélisation de l'incertitude dans les mesures grâce à une représentation des primitives par variables aléatoires.

Les environnements d'intérieur sont bien adaptés à ce type de modèle. En effet, ils sont en général composés essentiellement de parties auxquelles on peut attacher des attributs géométriques de base (segments de droites, plans,...). Pour être à la fois fiable et concis, le modèle devra comporter les composants dont les attributs n'évoluent pas dans le temps tels que les murs, portes, fenêtres, ou ceux dont la modification des attributs ne peut être due au fait du hasard tel que les gros meubles.

Le robot se déplace dans un plan (espace à deux dimensions) délimité par des surfaces majoritairement verticales. Il perçoit les arêtes délimitant ces surfaces et les utilise comme amers pour la localisation. Le modèle doit donc comprendre une description de ces arêtes ainsi que leur position et orientation. De plus, les occultations mutuelles de ces surfaces doivent également être prises en compte.

## Modélisation des primitives 3D

### Quelques représentations d'objets 3D

Il existe plusieurs types de représentations pour des objets à trois dimensions. Ce sont principalement les représentations filaires, les représentations surfaciques et les représentations volumiques ([Shaheen99]).

#### La représentation filaire

Historiquement c'est la première représentation graphique d'objets 3D à avoir été mise en œuvre. Le modèle est en fait un ensemble de segments correspondant aux arêtes et aux frontières de certaines surfaces de l'environnement. Le modèle retient les coordonnées des sommets et les liaisons entre ces sommets définissant les arêtes. La représentation en mémoire d'un tel modèle se résume donc à une liste de couples de sommets définis par leurs coordonnées dans un repère 3D (figure 2.4).

La capacité mémoire nécessaire est donc très faible. La mise à jour lors d'éventuels ajouts ou de déplacements d'éléments de l'environnement est également très facile et rapide.

Ce type de modèles présente toutefois une faiblesse. En effet, la représentation filaire est en quelques sortes transparente. Elle ne tient pas compte des occultations mutuelles entre les faces de l'environnement. Il en découle que plusieurs interprétations d'une même vue peuvent être retenues.

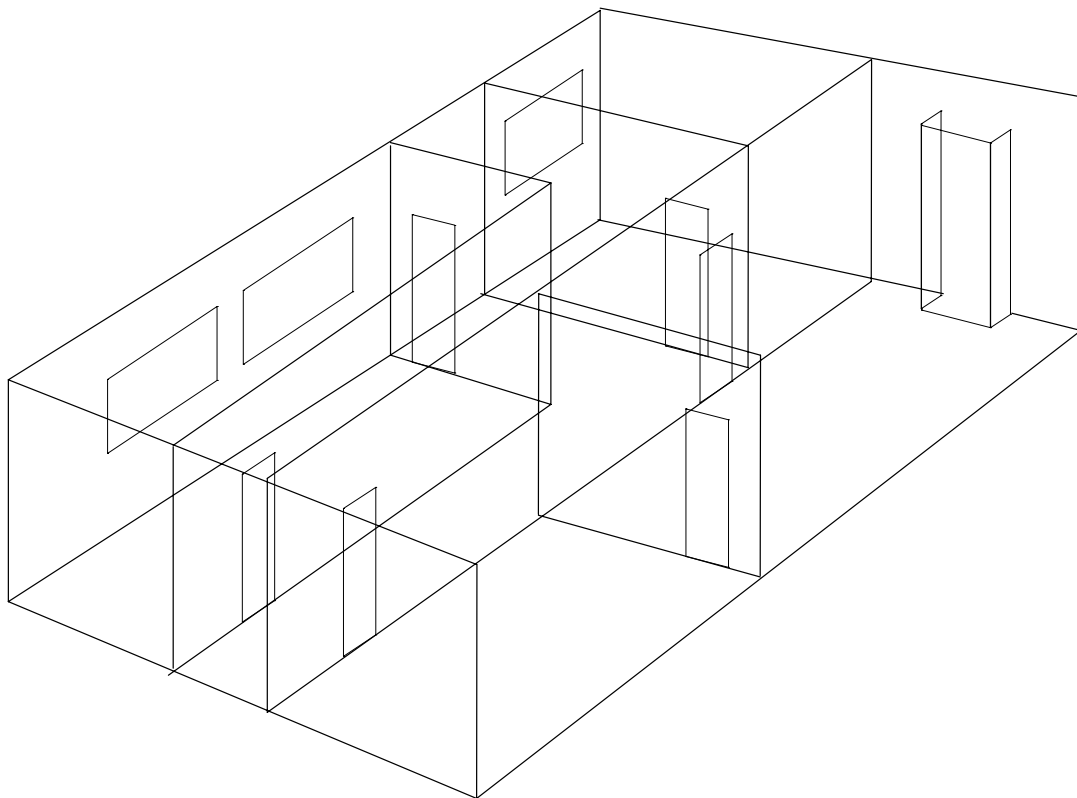
## La représentation surfacique

Les objets sont représentés grâce à leurs surfaces externes. Cette surface peut être définie par des éléments de base tels que les plans ou les surfaces de révolution et des liens topologiques entre ces surfaces. On parle alors de BREP (ou Boundary REPresentation) ([Qiang89]). Elle peuvent également être définies par leur ensemble de vecteurs normaux répartis sur une sphère dans laquelle est centré l'objet. C'est l'Image Gaussienne Etendue (IGE) ([Ikeuchi88]). Les volumes peuvent être définis avec cette représentation si l'on spécifie les lieux de la matière.

## La représentation volumique

Une représentation possible est la Constructive Solid Geometry (CSG) dans laquelle les volumes sont constitués de blocs élémentaires tels que les sphères, le cylindres ou les polyèdres. L'assemblage se fait grâce à des opérateurs arithmétiques ( $\cap, \cup, -, \dots$ ). Une autre représentation est l'énumération spatiale où la scène est décomposée en *voxels* de taille fixe auxquels sont associés un état correspondant à la présence ou non de la matière ([Pampagnin90]).

Les primitives choisies étant les segments de droites, le modèle fil de fer semble le plus indiqué. Il est le plus simple à mettre en œuvre mais ne tient pas compte des occultations. Ce problème est généralement géré en vision par ordinateur par la construction d'un graphe d'aspects ([Bowyer92]). Dans le contexte d'un robot mobile on parle plutôt de régions d'invariance visuelle ([Simsarian98], [Talluri96]). Un plan de l'espace 2D navigable du robot décomposé en telles régions est donc associé au modèle fil de fer.



*Figure 2.4 : exemple de modèle fil de fer d'un environnement intérieur*

## Régions d'invariance visuelle

La notion de régions d'invariance visuelle est similaire à celle de graphe d'aspect utilisée en reconnaissance d'objets. Elle a été définie pour la robotique mobile par Guibas ([Guibas92]). Elle a été également utilisée dans ce cadre par Talluri ([Talluri96]) pour un environnement composé uniquement de segments horizontaux, puis par Simsarian ([Simsarian98]). L'espace navigable du robot (ou l'espace de déplacement de la caméra) est découpé en régions. A chaque région est associée une liste (sous-ensemble) de primitives du modèle global de l'environnement. La connaissance, même très approximative, d'une estimation de la position peut alors servir (comme nous le verrons dans la partie mise en correspondance) à restreindre la recherche d'appariements à ces primitives ou, dans le pire des cas, à l'ensemble de primitives formé par l'union des listes associées à quelques régions seulement. Le calcul de ces régions étant effectué hors ligne, il est évident que le gain en temps de calcul en ligne peut être considérable.

La liste de segments visibles à partir d'une région donnée est liée aux occultations mutuelles des surface composant l'environnement (délimitées par les segments du modèle) mais aussi à l'orientation de la caméra dans cette région. Dans ce qui suit nous présentons quelques définitions rigoureuses sur les régions d'invariance visuelle puis la méthode de leur construction.

### Définitions

Soit un environnement modélisé grâce à une description fil de fer par un ensemble de segments 3D. Une partie du plan de l'espace horizontal 2D, délimitée par un polygone global, est associée à ce modèle. Le robot (et donc le centre optique de la caméra), se déplace à l'intérieur de ce polygone. La définition de régions d'invariance visuelle est basée sur la notion de visibilité ([Simsarian98]). Nous allons donc rappeler quelques définitions permettant d'aboutir à la construction de ces régions.

#### *Définition 1*

Un point  $p_1$  est visible pour un point  $p_2$  si le segment  $p_1 p_2$  est intérieur au polygone global contenant  $p_1$  et  $p_2$ .

#### *Définition 2*

Un segment  $L$  est visible pour un point  $p_2$  s'il existe au moins un point  $p_1$  sur ce segment tel que  $p_1$  est visible pour  $p_2$ .

#### *Définition 3*

On appelle liste de visibilité  $L_p$ , d'un point  $p$ , l'ensemble des segments visibles pour ce point.

#### *Définition 4*

Une région d'invariance visuelle  $RIV$  est un polygone  $P$  inclus dans le polygone global et dont tous les points  $p$  possèdent la même liste de visibilité  $L_p$ .

Une propriété des régions d'invariance visuelle est qu'elles sont disjointes et leur union forme le polygone global ([Simsarian98]).

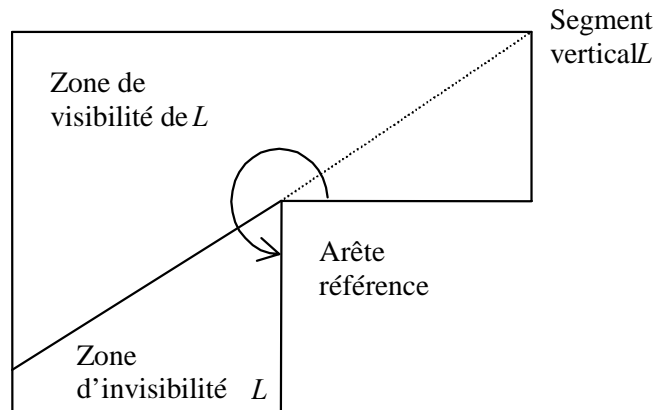
### Méthode de construction

La première étape dans la construction des régions d'invariance visuelle est de trouver les droites frontières. Le franchissement de chacune de ces droites provoque l'apparition ou l'occultation d'une primitive. Dans un environnement intérieur classique, le phénomène

d'occultation est le plus souvent lié aux arêtes verticales du modèle où le polygone global est convexe.

La construction des frontières se fait alors de la façon suivante. Pour chaque arête convexe du modèle, que l'on désignera par arête de référence, effectuer un balayage circulaire dont l'axe est cette arête. A chaque fois qu'un segment du modèle (ou une extrémité de segment si ce dernier n'est pas vertical) est rencontré, tirer la droite reliant ce segment ou cette extrémité de segment à l'arête référence. La partie de cette droite qui est du côté opposé au segment (ou à l'extrémité de segment) par rapport à l'arête référence et qui est contenue dans le polygone global est une frontière de région d'invariance.

L'intersection de toutes les droites ainsi tracées après exécution de cette opération sur toutes les arêtes concaves forme l'ensemble des régions d'invariance visuelle (figure 2.5).



**Figure 2.5 :** construction de la frontière d'une région d'invariance visuelle

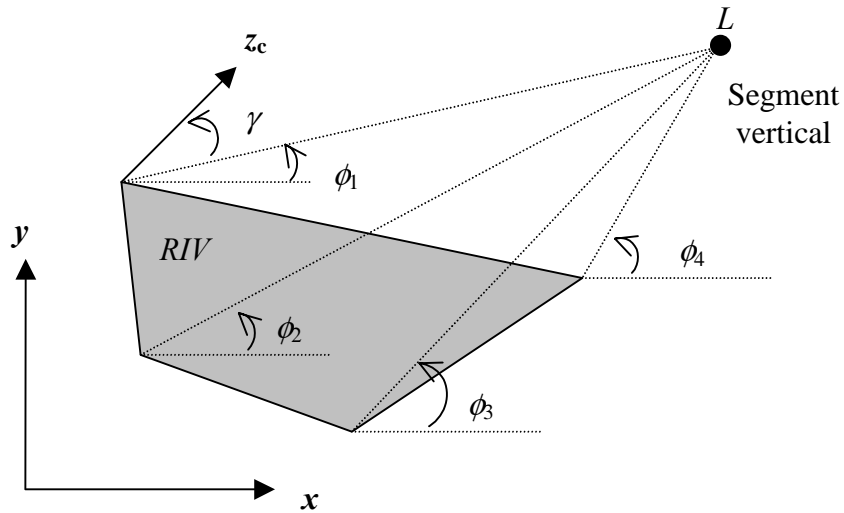
Après le calcul des frontières, les régions obtenues sont numérotées. A Chaque région  $RIV_i$  est attachée une liste de visibilité  $L_i$  qui est l'ensemble des segments du modèle visibles à partir de cette région.

## Gestion de l'orientation

La visibilité d'un segment dépend également de l'orientation de la caméra. Pour chaque segment de la liste de visibilité associée à une  $RIV$  donnée, un intervalle est déterminé pour l'orientation de la caméra à partir de laquelle il est visible.

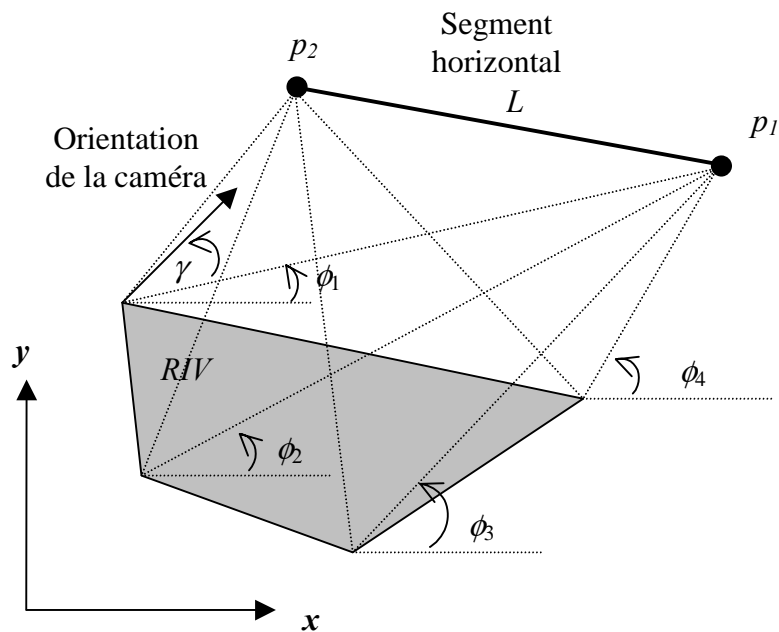
Dans les cas d'un segment vertical, l'orientation à partir de laquelle ce dernier peut être vu est en fait comprise entre  $\phi_{\min}$  et  $\phi_{\max}$  avec :

- $\phi_{\min}$  l'angle minimal formé par la droite reliant le segment observé à chaque sommet du polygone délimitant la  $RIV$  avec l'axe des abscisses, auquel on additionne l'angle de vue  $\gamma$  de la caméra.
- $\phi_{\max}$  l'angle maximal formé par la droite reliant le segment observé à chaque sommet du polygone délimitant la  $RIV$  auquel on retranche l'angle de vue  $\gamma$  la caméra (figure 2.6).



**Figure 2.6 :** limitation de l'angle de vue d'un segment vertical  $L$  du modèle dans une RIV

Dans le cas d'un segment non vertical, le calcul précédent est répété avec les coordonnées des projections de ses extrémités  $p_1$  et  $p_2$  sur le plan horizontal  $(x,y)$ . Deux intervalles d'orientation sont ainsi déterminés, une pour chaque extrémité :  $[\phi_{\min 1}, \phi_{\max 1}]$  et  $[\phi_{\min 2}, \phi_{\max 2}]$ . L'intervalle retenu est alors  $[\min(\phi_{\min 1}, \phi_{\min 2}), \max(\phi_{\max 1}, \phi_{\max 2})]$  (figure 2.7).



**Figure 2.7 :** limitation de l'angle de vue d'un segment horizontal  $L$  du modèle dans une RIV



# Extraction des primitives de l'image

Une image contient un très grand nombre de données représentées sous forme brute (pixels). Une grande partie de ces données ne comporte pas d'information pertinente. Dans le cadre d'une application en robotique (notamment en temps réel), il est indispensable de pouvoir identifier les fractions de l'image qui véhiculent des informations pertinentes en fonction du type de primitives du modèle que l'on cherche à identifier dans la scène. Il est également nécessaire de représenter ces informations de façon concise dans la mémoire du système.

La segmentation de l'image est à la base de tout système de vision. Elle a pour but de détecter et de caractériser par des attributs les zones d'intérêts de l'image. Il s'agit essentiellement des contours et des régions. Dans une image représentant une scène réelle avec un certain nombre d'entités, les contours correspondent généralement aux frontières et arêtes des objets, et les régions à leurs surfaces.

La phase de détection fournit comme résultat des ensembles de pixels de l'image appartenant à ces zones d'intérêt (contour ou région). L'objectif final étant de retrouver des objets de l'environnement, il faut définir ces ensembles par des représentations géométriques (droites, coniques, repères) manipulables et dont les caractéristiques sont comparables à celles des entités physiques auxquelles elles correspondent.

Bien que le sujet de la présente thèse ne soit pas le traitement d'image nous présentons dans ce qui suit les outils utilisés pour extraire les primitives géométriques (segments de droite) de l'image.

## Détection des contours

### Rappels

Le principe est de détecter les discontinuités d'ordre zéro des niveaux de gris. Deux approches existent pour cela. Il s'agit de l'approche gradient et de l'approche Laplacien. L'approche gradient consiste à détecter les pixels de l'image correspondant aux extréma locaux du gradient. L'approche Laplacien consiste à détecter les passages par zéro du Laplacien (figure 2.8).

### Gradient et Laplacien

Soit  $I(x,y)$  une image en niveaux de gris. Le Gradient de  $I$  en tout point  $I(x,y)$  est défini par sa norme

$$G(x,y) = \sqrt{(G_x(x,y))^2 + (G_y(x,y))^2} \quad (2.3.1)$$

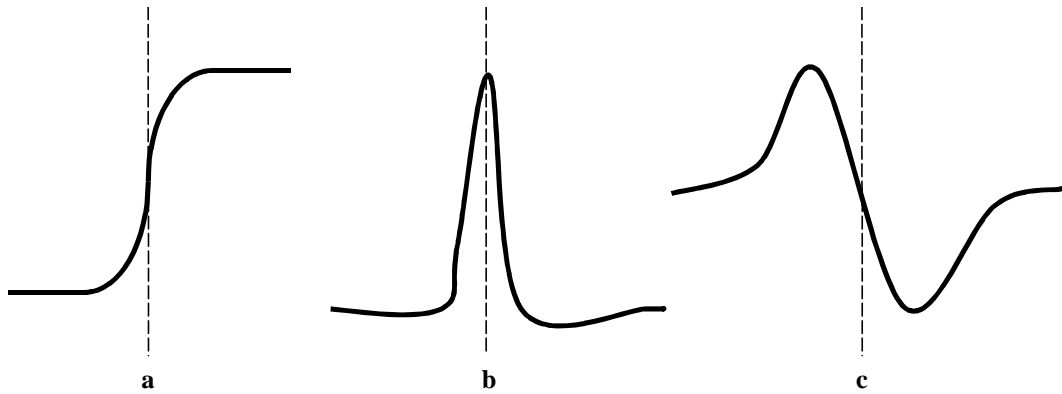
avec  $G_x(x,y) = \frac{\partial I(x,y)}{\partial x}$   $G_y(x,y) = \frac{\partial I(x,y)}{\partial y}$

et sa direction

$$\phi(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad (2.3.2)$$

Le Laplacien de l'image  $I$  est défini par

$$L(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} \quad (2.3.3)$$



**Figure 2.8** : détection de contours par l'approche gradient et l'approche Laplacien  
*a-discontinuité du signal b-gradient c-Laplacien*

### Filtrage linéaire

Le Filtrage d'une image  $I(x,y)$  définie dans le domaine  $\left[-\frac{N}{2}, +\frac{N}{2}\right]$  est le calcul de son produit de convolution avec une fonction  $f(x,y)$  définie dans le domaine  $\left[-\frac{K}{2}, +\frac{K}{2}\right]$  (avec  $K \leq N$ )

$$I_f = (I * f)(x, y) = \sum_{i'=-\frac{K}{2}}^{\frac{K}{2}} \sum_{j'=-\frac{K}{2}}^{\frac{K}{2}} f(i-i', j-j') I(i', j') \quad (2.3.4)$$

En choisissant convenablement les coefficients de  $f$  le résultat de ce filtrage peut être un lissage du bruit des niveaux de gris de l'image  $I$ .

### Détection de contour

En pratique la détection de contour se traduit par une opération de lissage puis de dérivation de l'image par convolution avec un filtre optimal ([Houard93]). De plus on a les deux propriétés suivantes

$$(I * f)' = I * f'' \quad (2.3.5)$$

$$(I * f)'' = I * f'' \quad (2.3.6)$$

Grâce à ces propriétés les dérivées première et seconde peuvent être estimées par convolution avec la dérivée première ou seconde d'un filtre de lissage  $f(x,y)$ .

Certains filtres 2D sont dits séparables. Un filtre  $f(x,y)$  est séparable si

$$f(x,y) = f_1(x) f_2(y) \quad (2.3.7)$$

Il s'en suit

$$I * f(x,y) = I * (f_1(x) f_2(y)) = I * f_1(x) * f_2(y) \quad (2.3.8)$$

Autrement dit, l'implantation d'un filtre séparable peut se faire grâce à la mise en cascade de deux filtres monodimensionnels. Ceci permet un gain évident en temps de calcul. Le gradient de l'image est finalement estimé par :

$$G_x(x, y) = \frac{\partial I(x, y)}{\partial x} = I(x, y) * (f'_1(x) f_2(y))$$

$$G_y(x, y) = \frac{\partial I(x, y)}{\partial y} = I(x, y) * (f_1(x) f'_2(y))$$
(2.3.9)

Dans le cas de l'approche Gradient, l'estimation du gradient ne suffit pas. Il faut ensuite déterminer les extréma locaux. Une liste des filtres optimaux les plus utilisés avec l'analyse de leurs performances ainsi que les méthodes de détections des extréma locaux se trouvent dans [Horaud93].

## Méthode utilisée

Dans ce travail nous appliquons d'abord un filtrage non-linéaire (filtre médian) sur l'image d'origine. Ceci a pour effet de lisser le bruit de façon plus efficace qu'avec un filtre linéaire. La méthode de Canny ([Horaud93]) est ensuite utilisée pour la détection des contours. Cette méthode est basée sur une approche Gradient. Le Gradient de l'image lissée est estimé grâce à la dérivée d'un filtre Gaussien  $f$ . Ce filtre étant séparable suivant les directions  $x$  et  $y$ , il est donc représenté en pratique par les deux filtres Gaussiens  $f_1$  et  $f_2$  suivants :

$$f_1(x) = e^{-\frac{1}{2} \frac{x^2}{\sigma^2}}, f_2(y) = e^{-\frac{1}{2} \frac{y^2}{\sigma^2}}$$
(2.3.11)

$$f_1'(x) = -\frac{x}{\sigma^2} e^{-\frac{1}{2} \frac{x^2}{\sigma^2}}, f_2'(y) = -\frac{y}{\sigma^2} e^{-\frac{1}{2} \frac{y^2}{\sigma^2}}$$
(2.3.12)

La largeur du filtre est déterminée automatiquement en fonction de la valeur utilisée pour  $\sigma$ . Pour une largeur maximale de 30 pixels, la valeur de la Gaussienne est calculée pour chaque pixels (de -30 à +30). Le filtre est tronqué lorsque cette valeur devient inférieure à un seuil (0,0001 dans notre cas).

La détection des extréma est effectuée grâce à deux seuils, un seuil haut pour les contours forts et un seuil bas (égal à 0,4 fois le seuil haut) pour les contours faibles. Les contours faibles sont inclus dans le résultat seulement s'ils sont connectés à un contour fort.

## Sélection des contours droits et approximation polygonale

Généralement, la phase qui suit la détection des contours est le chaînage. Cette opération consiste à regrouper les pixels de contours présents sous forme matricielle en un ensemble de listes chaînées. Chaque liste est ensuite assimilée à une entité géométrique (segment de droite, conique, etc) caractérisée par un nombre restreint de paramètres ou décomposée en plusieurs entités. L'implémentation du chaînage pose généralement un certain nombre de problèmes dont le temps de calcul et la représentation en mémoire des listes ([Horaud93], [Loaiza99], [Shaheen99]).

Dans notre cas, on recherche parmi les points de contours, les listes de pixels présentant un certain alignement pour les caractériser par un segment de droite. Plutôt que d'effectuer un chaînage classique nous adoptons plutôt une méthode de recherche dite globale basée sur la transformée de Hough. Le principe consiste à discrétiser l'espace formé par les paramètres d'une droite formant ainsi une table de cellules, chaque cellule représentant une droite. Pour chaque point de contour on vote pour toutes les cellules correspondant aux droites passant par ce point. L'accumulation des votes dans une cellule signifie l'existence d'un ensemble

important de pixels alignés selon cette direction. Il reste ensuite à vérifier la continuité de cet ensemble de pixels pour statuer sur la présence ou non d'un tel segment de droite dans l'image. Les étapes de cette technique sont détaillées dans les sections suivantes.

## Représentation des droites dans l'espace de Hough

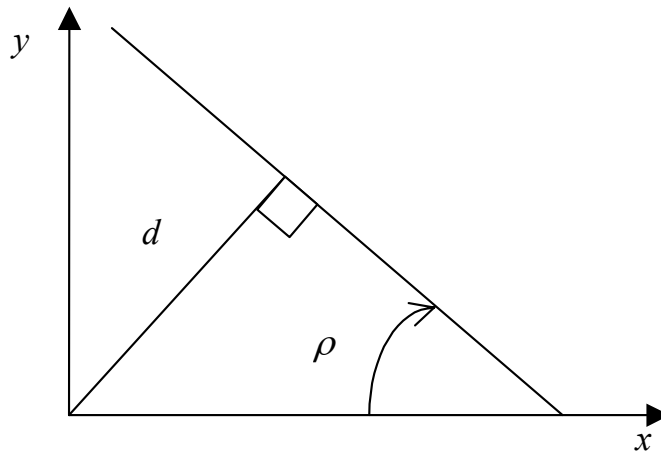
Une droite peut être caractérisée par son orientation  $\rho$  et sa distance à l'origine  $d$  (Figure 2.9) avec

- $\rho \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$  et  $d \in [0, d_{max}]$

- $d_{max}$  la moitié de la longueur de la diagonale de l'image.

Tout point  $(x,y)$  appartenant à une droite de paramètres  $(\rho,d)$  vérifie l'équation suivante

$$d = x.\cos(\theta) + y.\sin(\theta)$$



*Figure 2.9 : représentation d'une droite dans l'espace de Hough*

## Table de Hough

L'espace paramétrique à deux dimensions défini par  $\rho$  et  $d$  est discrétisé sous forme de tableau. Le nombre de cellules de ce tableau dépend des pas de discrétisation  $\Delta\rho$  et  $\Delta d$  de chacune des deux dimensions (figure 2.10). Les cellules sont numérotées grâce à deux indices  $i$  et  $j$  avec

- $i \in [-K, +K]$  où  $K = \text{arrondi}\left(\frac{d}{\Delta d}\right)$

- $j \in [-L, +L]$  où  $L = \text{arrondi}\left(\frac{\rho}{\Delta\rho}\right)$ .

La cellule  $(i,j)$  correspond à l'ensemble de droites dont les paramètres  $\rho$  et  $d$  appartiennent respectivement aux intervalles suivantes  $[i*\Delta\rho, (i+1)*\Delta\rho]$  et  $[j*\Delta d, (j+1)*\Delta d]$ .

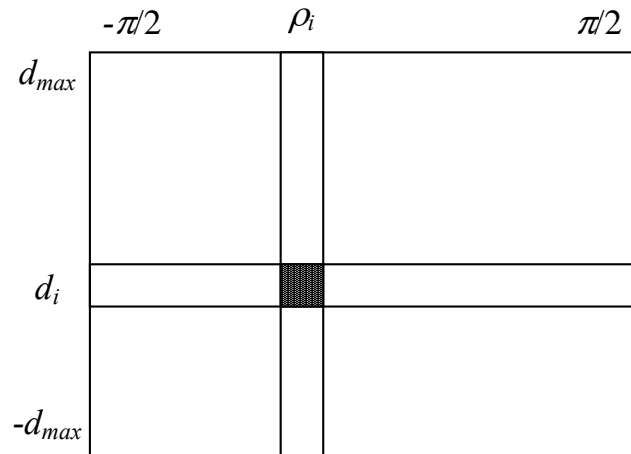


Figure 2.10 : table de Hough

## Détection des contours droits

La valeur de toutes les cellules de la table est initialisée à zéro. Un balayage de tous les points de contours est effectué. Pour chaque pixel  $(x,y)$ , toutes les cellules sont testées comme suit :  
 Pour  $j$  allant de  $-L$  à  $L$

$$i = \text{partie entière de} \left( \frac{(x \cdot \cos(j \cdot \Delta\rho) + y \cdot \sin(j \cdot \Delta\rho))}{\Delta d} \right)$$

incrémenter la cellule  $(i,j)$ ,

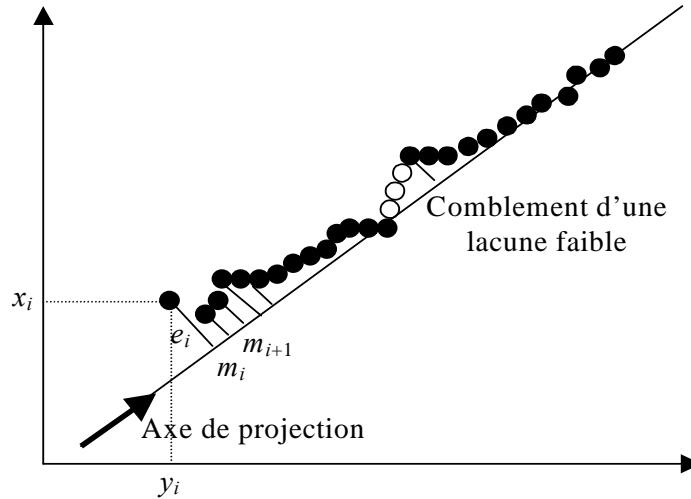
ajouter le point  $(x,y)$  à la liste correspondant à la cellule  $(i,j)$ .

Fin de Pour.

A la fin du balayage des points de contour il y aura accumulation des votes dans les cellules correspondant aux droites auxquelles un grand nombre de points de contour sont assez proches. En appliquant un seuil minimal sur la valeurs des cellules on peut retenir les listes de pixels qui constituent potentiellement des segments de droite sur l'image. Le seuil définit la longueur minimale des segments que l'on gardera en fin de traitement.

## Chaînage des pixels

Chaque liste retenue contient un certain nombre de pixels répartis dans l'image et disposés dans certaines limites selon une droite. Pour correspondre à une primitive de l'environnement réel il faut que le contour présente peu de lacunes (trous ou points de ruptures). Les lacunes faibles peuvent être comblées car correspondant généralement à une mauvaise détection ou à du bruit lors de l'acquisition de l'image. Pour cela il faut d'abord effectuer un chaînage des pixels constituant le contours. Les algorithmes de chaînage et de comblement des lacunes pour des contours de forme quelconque peuvent être coûteux en temps de calcul. La méthode proposée permet d'effectuer rapidement un chaînage et un comblement des lacunes en exploitant le fait que les contours étudiés sont des segments de droites.



**Figure 2.11** : chaînage des points de contours

Pour une liste de paramètres  $(\rho, d)$ , chaque pixel  $(x, y)$  de la liste est d'abord projeté sur la droite  $(\rho, d)$ . Les pixels sont alors ordonnés en fonction de leurs coordonnées sur cet axe unidimensionnel (figure 2.11). Les pixels étant très proches de la droite sur laquelle ils sont projetés, l'ordre de chaînage ainsi obtenu est généralement celui d'un chaînage classique. La liste est ensuite parcourue dans une direction donnée. Chaque fois que la distance entre deux pixels successifs est supérieure à un seuil donné, une nouvelle chaîne est formée avec les pixels déjà parcourus et une nouvelle liste est commencée à partir du pixel en cours.

## Approximation polygonale

Les paramètres  $\rho$  et  $d$  pour chaque liste ne sont qu'une estimation des vrais paramètres du contour droit détecté. Il faut donc à présent calculer les valeurs de ces paramètres qui caractérisent au mieux le contour, autrement dit, définir la droite analytique qui optimise les distances aux points composant le contour (figure 2.11). Ceci est fait par la méthode des moindres carrés.

Soit une liste de points de contour  $p_i(x_i, y_i)$ . La distance d'un point de cette liste à une droite de paramètres  $(\rho, d)$  s'écrit

$$e_i = |x_i \sin(\rho) + y_i \cos(\rho) - d| \quad (2.3.13)$$

On cherche donc les valeurs des paramètres  $(\rho, d)$  qui minimisent le critère suivant

$$E = \sum_i e_i^2$$

Les valeurs recherchées sont les solutions du système suivant

$$\begin{cases} \frac{\partial E}{\partial \rho} = 0 \\ \frac{\partial E}{\partial d} = 0 \end{cases}$$

Ce qui donne après calcul

$$d = V_x \sin(\rho) + V_y \cos(\rho) \quad (2.3.14)$$

$$\rho = \arctan\left(\frac{2V_{xy}}{V_{yy} - V_{xx}}\right) \quad (2.3.15)$$

avec

$$V_x = \frac{1}{n} \sum_{i=1}^n x_i \quad V_y = \frac{1}{n} \sum_{i=1}^n y_i$$

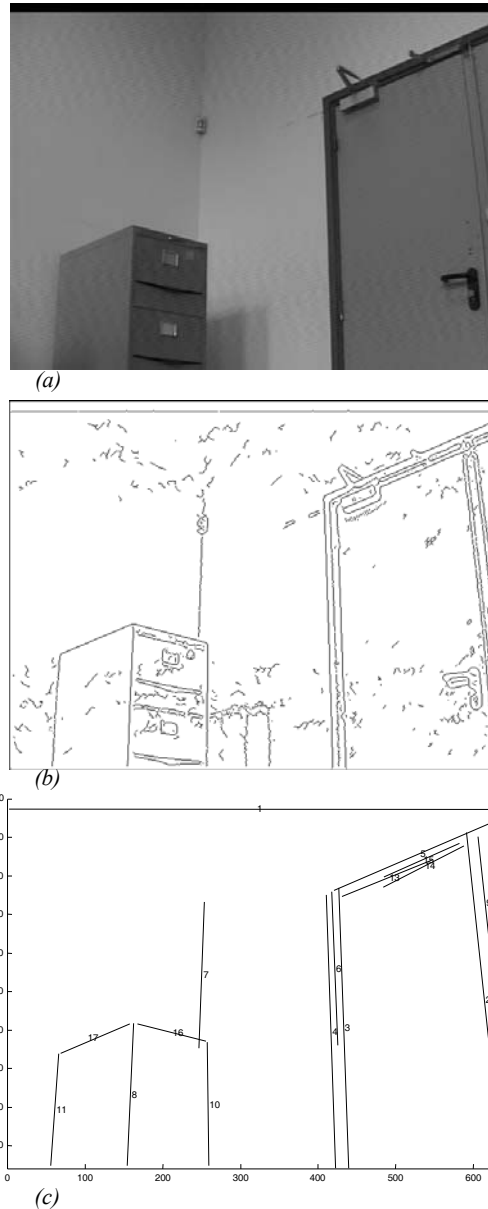
$$V_{xx} = \sum_{i=1}^n (x_i - V_x)^2 \quad V_{yy} = \sum_{i=1}^n (y_i - V_y)^2 \quad V_{xy} = \sum_{i=1}^n (x_i - V_x)(y_i - V_y)$$

## Problèmes liés à la transformée de Hough

Le principal problème rencontré lorsque l'on utilise une méthode basée sur la transformée de Hough (y compris dans d'autres domaines que la détection de primitives géométriques) est le pas de discrétisation. En effet un pas trop petit risque de rendre l'exécution trop lente, de même que certains segments peuvent ne pas être détectés à cause de la dispersion de leurs votes dans des cellules voisines. A l'inverse, un pas trop grand peut conduire à la fusion des votes correspondant en réalité à des segments dont les paramètres sont proches. Il y a alors apparition de fausses détections. Il faut donc réaliser un compromis entre ces deux effets ([Shaheen99]).

## Exemple de résultats

La figure 2.12 montre un exemple de résultat obtenu avec l'algorithme développé. En (a) on voit l'image d'origine d'une scène d'intérieur, en (b) l'image des contours obtenus par l'approche gradient et en (c) la liste des segments retenus après approximation polygonale et seuillage sur la longueur des segments.



**Figure 2.12** : résultat de l'extraction des primitives de la scène : (a) image originale, (b) contours, (c) segmentation et seuillage sur la longueur des segments

La figure montre sur un cas simple qu'avec un choix approprié de la valeur du seuil, outre l'élimination des bruits, on réduit considérablement la quantité de données correspondant à des objets non modélisables de l'environnement. Cette étape est donc déterminante sur la suite du processus de localisation, notamment sur la phase de mise en correspondance.



# Calcul de la position et de l'orientation de la caméra

Ce chapitre concerne la localisation proprement dite de la caméra, autrement dit le calcul de sa position et de son orientation relativement au modèle. Ce calcul est basé sur un certain nombre d'appariements entre les primitives du modèle et celles de l'image. Dans ce chapitre nous supposons que l'on dispose de ces appariements (les méthodes pour obtenir sont présentées dans le chapitre suivant).

Nous commencerons par un état de l'art sur les techniques existantes de localisation de caméra grâce à des appariements 2D-3D. Nous discuterons ces méthodes en fonction des techniques de calcul et du type de primitives utilisées. Nous nous intéresserons ensuite aux méthodes que nous avons développées. Nous présenterons deux approches numériques de localisation grâce à des appariements de segments de droites. Il s'agit de l'approche de Lowe et de l'approche de Phong-Horaud. Nous aborderons leurs adaptations au contexte de la localisation d'un robot mobile en vue d'optimiser leurs performances dans ce cadre. Des résultats expérimentaux seront également présentés en fin de chapitre.

## Etat de l'art

Le calcul de la position et de l'orientation de la caméra relativement à une scène, parfois appelé «calibrage extrinsèque de caméra», est un problème central en vision par ordinateur. Il est utilisé notamment en navigation visuelle, coordination entre manipulateur et caméra, reconnaissance de formes, réalité augmentée, réalité virtuelle, etc.

Lorsque ce calcul utilise une connaissance *a priori* des objets constituant la scène filmée, on parle de localisation référencée modèle. Le modèle en question consiste en une description géométrique des objets de la scène. La disponibilité d'un certain nombre de correspondances entre les éléments de l'image et les parties du modèle ainsi que la connaissance des paramètres intrinsèques de la caméra (par calibrage) sont utilisées pour dériver un certain nombre d'équations permettant de recouvrir la position et l'orientation à partir de laquelle a été prise la vue.

Compte tenu de la diversité des domaines d'application de cette approche et des exigences particulières de chaque domaine, il existe une grande variété de techniques basées sur le modèle pour la localisation de caméra. Le choix d'une méthode peut être motivé par la rapidité d'exécution, la précision du résultat, la robustesse en présence de données bruitées ou de conditions initiales variables, etc.

Ces méthodes peuvent être regroupées selon différents critères. Premièrement, du point de vue de la formulation mathématique du problème, on peut distinguer deux catégories : les méthodes analytiques et les méthodes numériques ([Carceroni97]). Ensuite, il y a le type de primitives utilisées pour les correspondances 2D-3D. En effet, les méthodes peuvent être basées sur des correspondances de points, de droites, de courbes ou encore d'angles. Enfin, le choix du modèle interne intervient également puisque certaines méthodes utilisent soit des modèles en pleine perspective soit des modèles simplifiés (perspective faible ou para-perspective) alors que d'autres combinent deux modèles.

Les méthodes analytiques sont basées sur la résolution analytique du système d'équations obtenues grâce à chaque appariement 3D-2D. Elles utilisent un nombre fixe d'équations (généralement égal au nombre d'inconnues composant l'orientation et la position à calculer). Les équations supplémentaires sont parfois utilisées pour lever les ambiguïtés dues à la multiplicité des solutions possibles. Les méthodes numériques ont pour principe l'optimisation itérative d'une fonction d'erreur exprimée en fonction des six paramètres de localisation composant l'orientation et la position de la caméra. Plus le nombre de correspondances 2D-3D (donc d'équations) est élevé mieux le système est conditionné (surdéterminé). Plus adaptées au problème étudié vu la nature bruitée des mesures (primitives extraites sur l'image), elles présentent néanmoins l'inconvénient de nécessiter pour certaines une estimation initiale de la localisation pour converger.

D'un autre point de vue, on peut également distinguer principalement deux grandes familles qui regroupent d'une part les méthodes basées sur des correspondances de points et d'autres part les méthodes utilisant des correspondances de droites. Les droites forment des sources de primitives plus robustes car elles peuvent être extraites d'une image de manière plus fiable. En outre, un ensemble de correspondances de points peut être transformé en un ensemble de correspondances de droites en reliant des points deux à deux (l'inverse n'est pas toujours possible).

Le modèle interne de caméra utilisé peut aussi différer d'une méthode à une autre. En effet, lorsque les dimensions des objets de la scène sont petites par rapport à leur distance à l'objectif l'utilisation du modèle à faible ou para-perspective permet de simplifier les calculs. En revanche, si ces objets sont de grande dimension, les effets de la perspective doivent être pris en compte soit en adoptant un modèle de projection perspective soit en combinant les

deux types de modèles, l'approximation affine étant généralement utilisée pour l'estimation initiale des paramètres de position et d'orientation.

Dans ce qui suit, un état de l'art en matière de localisation de caméra basée sur le modèle est dressé. Les principales méthodes de chaque catégorie sont présentées pour illustrer les différentes évolutions survenues dans le domaine.

## Formalisation du problème de la localisation de caméra

Un point commun à toutes les méthodes évoquées plus haut est qu'elles considèrent toutes au moins deux repères ou systèmes de coordonnées, le premier lié au modèle et le second lié à la caméra généralement défini comme dans la section 2.1. Localiser la caméra revient dans tous les cas à calculer la translation et la rotation qui relie les deux repères.

Le choix du formalisme mathématique pour définir cette transformation est alors déterminant pour les performances de chaque méthode. La représentation la plus naturelle définit la rotation par ses angles d'Euler ou les angles de roulis, tangage et lacet et la translation par un vecteur de l'espace ([Dombre88]). Cette représentation a pour avantage de minimiser le nombre de variables (pas de redondance). La rotation donne, cependant, des équations non-linéaires et présente des singularités et des ambiguïtés. Certains auteurs ont préféré la représentations par les quaternions unitaires ([Dombre88]). D'autres encore supposent les composantes de la matrice de rotation indépendantes ce qui a pour but de linéariser les expressions ([Liu90]).

Certaines approches incorporent des paramètres intrinsèques tel que la distance focale parmi les variables ([Lowe87]). Il existe aussi des méthodes avec des caméras non calibrées ([Quan97]).

## Méthodes analytiques

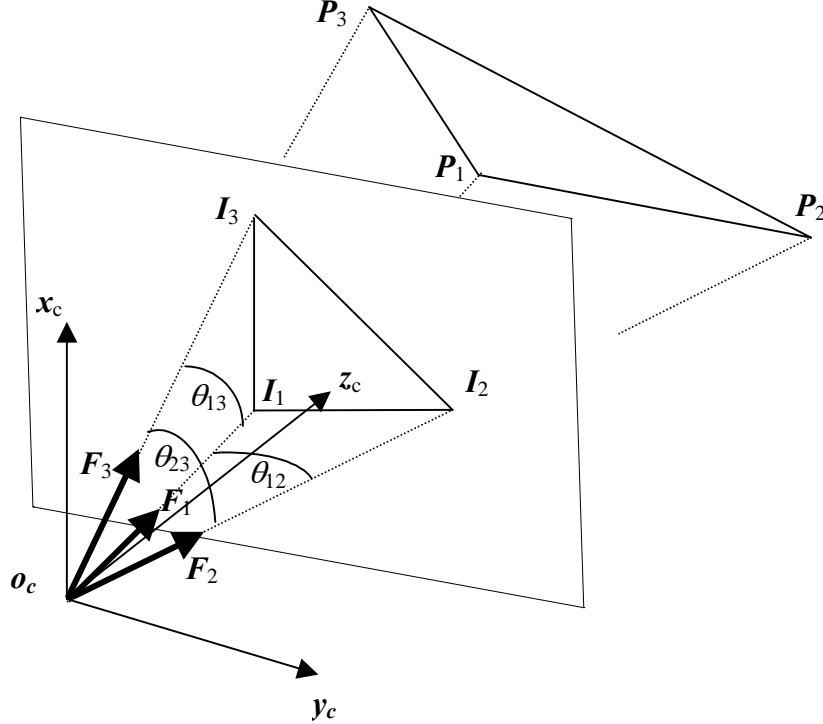
La mise en correspondance d'une primitive 2D (de l'image) avec une primitive 3D (du modèle) peut être traduite par une équation analytique dont les variables sont les paramètres de position et d'orientation définissant la transformation entre les deux repères objet et caméra. Il faut donc généralement autant de correspondances que d'inconnues pour résoudre le système d'équations obtenu. En général, six paramètres sont considérés pour définir cette transformation, trois pour la rotation et trois pour la translation.

## Correspondances de points

L'une des premières solutions fut celle de Fischler et Bolles ([Fischler81]) qui introduisirent le terme «perspective- $n$ -points» ( $PnP$ ) pour désigner le problème de la localisation d'un objet rigide par rapport à une caméra en utilisant  $n$  points de correspondance 3D-2D. Chaque correspondance impose deux contraintes (une sur l'axe des  $u$ , l'autre sur l'axe des  $v$ ). Ils obtiennent ainsi un système de  $n$  équations polynomiales. Il faut donc un minimum de trois correspondances de ce type pour recouvrir les six inconnues du système dans le cas général. Dans le cas du problème P3P, on dispose de trois points du modèle  $P_1, P_2, P_3$  ainsi que de leur pixels de projections respectives  $I_1, I_2, I_3$  (figure 3.1). Les angles  $\theta_1, \theta_2, \theta_3$  entre les faisceaux visuels  $F_1, F_2, F_3$  partant du centre optique et allant vers les trois points du modèle, en passant par leurs projections sur le plan image, sont calculés à partir des données image uniquement. Les distances  $l_{12}, l_{13}, l_{23}$  entre les trois points modèles sont connues. En utilisant la loi du cosinus un système d'équations polynomiales dont les inconnues sont les distances  $d_1, d_2, d_3$  du centre optique à chaque point modèle est obtenu

$$\begin{cases} d_1^2 + d_2^2 - 2.d_1.d_2.\cos(\theta_{12}) = l_{12}^2 \\ d_1^2 + d_3^2 - 2.d_1.d_3.\cos(\theta_{13}) = l_{13}^2 \\ d_2^2 + d_3^2 - 2.d_2.d_3.\cos(\theta_{23}) = l_{23}^2 \end{cases} \quad (3.1.1)$$

La position et l'orientation peuvent alors facilement être définies une fois ces distances calculées. Les auteurs établissent qu'avec 3 points d'appariement, le problème peut avoir théoriquement jusqu'à 8 solutions. D'un point de vue physique, ce nombre peut être ramené à 4 points de vue cohérents avec l'image.



**Figure 3.1** : méthode analytique à trois points

Abidi et Chandra [Abidi95] ont proposé une méthode pour calculer une solution unique avec quatre correspondances de points coplanaires  $P_1, P_2, P_3, P_4$ . Leur méthode inclut également le calcul de la distance focale  $f$ . Chaque triplet de points forme avec le centre optique un trièdre dont le volume peut être calculé par deux formules différentes qui sont

$$V = \frac{h.A}{3} \quad (3.1.2)$$

et

$$V = \mathbf{P}_1 \cdot \mathbf{P}_2 \wedge \mathbf{P}_3 \quad (3.1.3)$$

où  $h$  est la hauteur du trièdre et les vecteurs  $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$  décrivent les points modèle dans le repère caméra. Les vecteurs unitaires de  $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$  peuvent être exprimés en fonctions de données images seulement. C'est-à-dire en fonction des pixels de projection de chaque point du modèle et de la distance focale  $f$ . En mettant en équation les deux expressions de  $V$  pour chaque trièdre,  $h$  est exprimé en fonction des distances  $d_1, d_2, d_3, d_4$  de chaque point du modèle au centre optique et de la distance focale. Comme pour la méthode précédente, la connaissance de ces distances revient à localiser la caméra relativement au plan modèle formé par les quatre points. En mettant en correspondance les équations ainsi obtenues pour chaque trièdre, chaque triplet de distances  $d_i$  peut être exprimé en fonction de la quatrième distance (mettons  $d_1$ ) et de la distance focale réduisant ainsi le nombre d'inconnues du problème à

deux. Finalement, la distance  $P_1P_2$  peut être exprimée en fonction de  $d_1$  et  $f$ , de même que la distance  $P_1P_3$ . L'expression du rapport (dont la valeur est connue) de ces deux distances, après simplification du terme  $d_1$ , permet de calculer la distance focale  $f$ . Les autres inconnues sont alors calculées par remplacement.

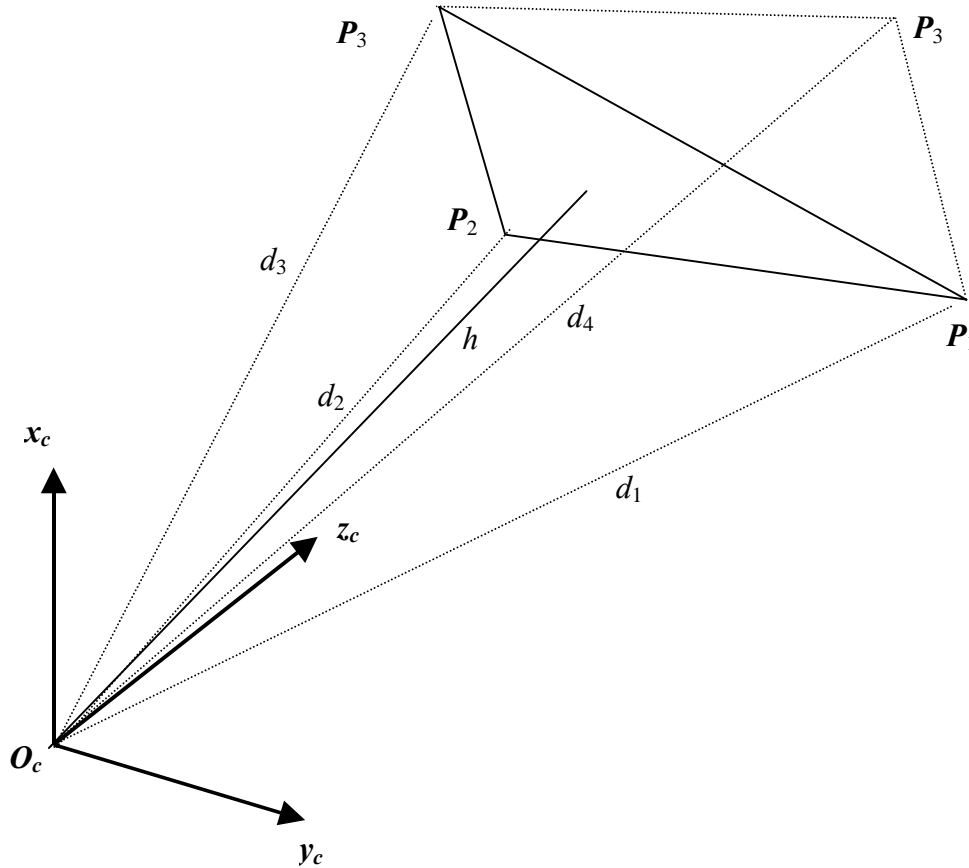


Figure 3.2 : méthode analytique à quatre trois points coplanaires

Quan [Quan99] présente une famille de méthodes linéaires pour calculer une solution unique en utilisant quatre ou plus points de correspondance. Les méthodes sont plus générales, du point de vue de la configuration des points du modèle, que celle de Abidi et Chandra et ne présentent pas de singularité lorsque les points sont coplanaires. Toutefois, la caméra est supposée calibrée (on connaît la matrice des paramètres de calibrage).

Soient  $N$  points du modèle  $P_1, P_2, P_3, \dots, P_N$  (avec  $N \geq 4$ ) dont les projections sur l'image sont connues. Comme pour les méthodes précédentes, Chaque couple  $P_i, P_j$  de points fournit une équation dérivant de la loi du cosinus qui peut s'écrire sous la forme de la contrainte suivante :

$$f_{ij}(x_i, x_j) = x_i^2 + x_j^2 - 2 \cdot x_i \cdot x_j \cdot \cos(\theta_{ij}) - l_{ij}^2 = 0 \quad (3.1.4)$$

où  $x_i$  représente la distance du centre optique au  $i^{\text{ème}}$  point de correspondance.

Cette expression ne contient pas de terme linéaire et la transformation  $x_i \rightarrow -x_i$  préserve la forme de l'expression. Les huit solutions possibles sont donc quatre paires de solutions symétriques. Chaque triplet d'équations (3.1.4) permet par remplacements successifs d'écrire

une équation polynomiale de degré 4 à une seule inconnue  $x$  qui est le carré de l'une des variables  $x_i$ , par exemple  $x_1$ , et qui s'écrit sous la forme :

$$g(x) = a_5x^4 + a_4x^3 + a_3x^2 + a_2x + a_1 = 0 \quad (3.1.5)$$

Avec  $N$  points de correspondance nous obtenons un système à  $K = \frac{(N-1)(N-2)}{2}$  équations de la forme (3.1.5) qui peuvent se mettre sous la forme matricielle suivante :

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & a_4^{(1)} & a_5^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1^{(K)} & a_2^{(K)} & a_3^{(K)} & a_4^{(K)} & a_5^{(K)} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{pmatrix} = \mathbf{A}_{K \times 5} \mathbf{t}_5 = 0 \quad (3.4)$$

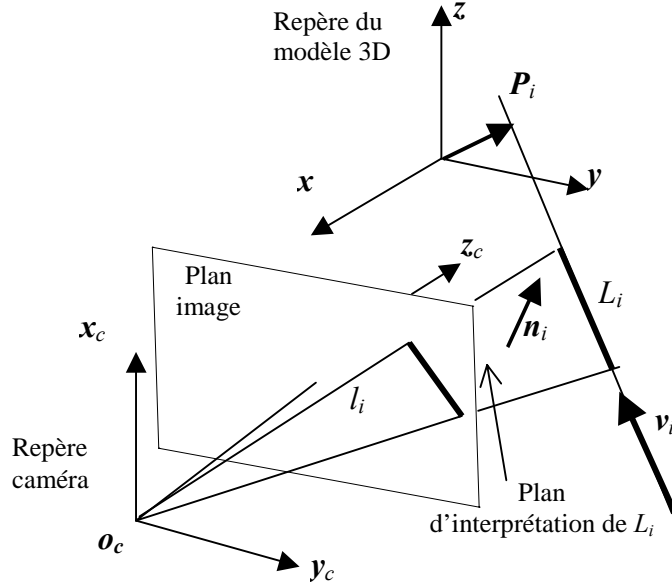
Lorsque  $N > 4$  cette équation peut être résolue grâce à la décomposition en valeurs singulières (SVD) de la matrice  $\mathbf{A}_{K \times 5}$  qui a alors un rang suffisamment élevé. Pour le cas où  $N=4$ , le calcul est plus complexe et nécessite deux SVD successives et l'exploitation de la propriété suivante du vecteur  $\mathbf{t}_5$

$$\mathbf{t}_5(i)\mathbf{t}_5(j) = \mathbf{t}_5(k)\mathbf{t}_5(l) \quad (3.1.6)$$

pour  $i+j = k+l$  et  $0 \leq i, j, k, l \leq 4$ .

## Correspondances de droites

Horaud présente une solution basée sur des correspondances de droites ([Horaud89]). Sa solution utilise trois segments non-coplanaires qui se coupent en un même point (faisceau de segments). Les auteurs présentent cette méthode comme une solution générale du problème P4P puisque les segments utilisés sont constitués à partir de quatre points. Carceroni la classe plutôt comme une solution particulière du problème à trois correspondances de droites (la particularité étant l'existence d'un point unique d'intersection) et ce du fait que seule la connaissance du point d'intersection est nécessaire ([carceroni97]). L'originalité de son approche consiste en l'introduction de la notion de plan d'interprétation qui est le plan contenant le segment de droite du modèle, la droite de l'image et le centre optique (figure 3.3). Le vecteur normal à ce plan peut être calculé à partir des données image uniquement. Le problème consiste à rechercher la transformation qui fait que chaque segment du modèle appartient à un plan orthogonal à ces vecteurs.



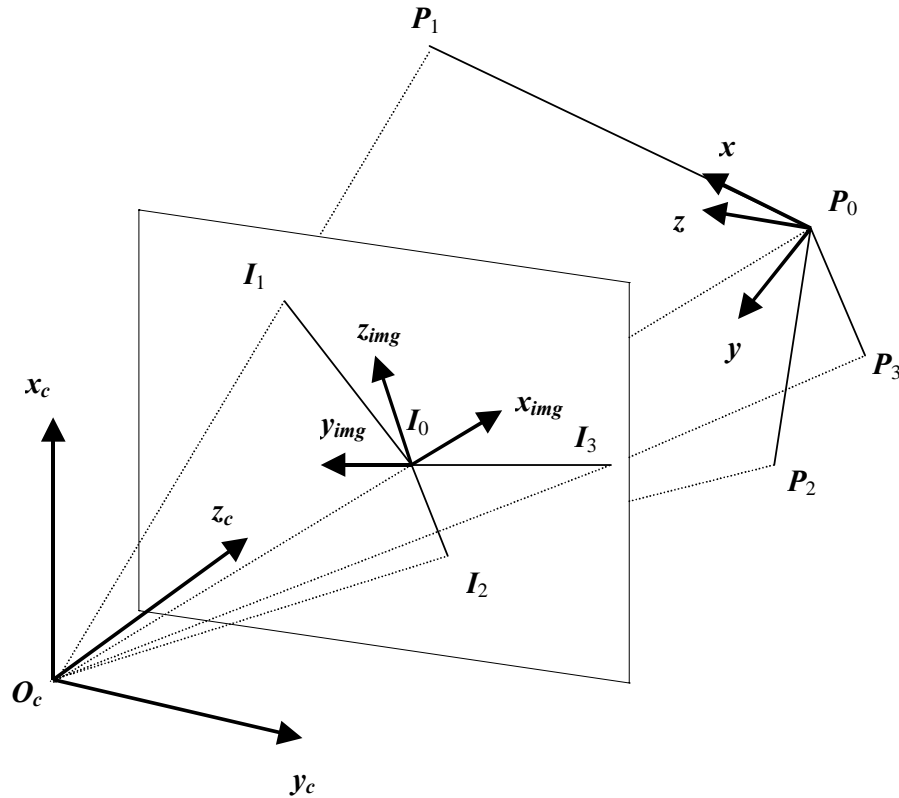
**Figure 3.3:** projection d'un segment 3D

Soient  $P_0, P_1, P_2, P_3$  quatre points du modèle 3D. Soient  $n_1, n_2, n_3$  les trois vecteurs, définis dans le repère-caméra ( $O_c, x, y, z$ ), normaux aux plans d'interprétation respectifs des segments  $P_0P_1, P_0P_2, P_0P_3$ . Le repère-modèle ( $O, x, y, z$ ) est choisi de telle sorte que l'axe  $x$  soit confondu avec un segment référence du modèle. Les auteurs introduisent, entre ces deux repères un repère intermédiaire dit repère-image ( $I_0, x_{img}, y_{img}, z_{img}$ ) défini comme suit : l'origine  $I_0$  est la projection de  $P_0$  sur le plan image, l'axe  $x_{img}$  est parallèle à  $O_cP_0$ , l'axe  $y_{img}$  est normal au plan d'interprétation du segment référence (figure 3.4). La position et l'orientation de la caméra sont définies par la composition des deux transformations caméra-image et image-modèle. La transformation caméra image est complètement indépendante du modèle et peut être calculée à partir de données image uniquement. La transformation image-modèle est composée d'une translation  $\overrightarrow{I_0P_0}$  et d'une rotation elle-même formée de deux rotations élémentaires, une première  $\theta$  autour de l'axe  $y_{img}$  ramenant l'axe  $x_{img}$  dans une position parallèle à  $x$  et une seconde  $\psi$  autour de l'axe  $x_{img}$  ramenant les axes  $y_{img}$  et  $z_{img}$  dans des positions parallèles aux axes  $y$  et  $z$ .

Deux contraintes imposant aux deux autres segments du modèle d'appartenir aux plans orthogonaux aux vecteurs normaux calculés à partir de l'image s'écrivent alors

$$\begin{aligned} \overrightarrow{P_0P_2} \cdot n_2 &= 0 \\ \overrightarrow{P_0P_3} \cdot n_3 &= 0 \end{aligned} \quad (3.1.7)$$

En écrivant les vecteurs  $\overrightarrow{P_0P_2}$  et  $\overrightarrow{P_0P_3}$  en fonction des deux angles de rotations, le système d'équation produit une équation polynomiale de degrés 4 et d'inconnue  $\cos(\theta)$ . Une fois la rotation connue, le calcul de la translation devient trivial.



*Figure 3.4 : repère-caméra, repère-image et repère-modèle*

Une solution plus générale est présentée par Dhome ([Dhome89]). Elle est basée sur 3 correspondances de droites quelconques. Dans cette méthode également, un repère-image est introduit entre les deux repères classiques. Cette fois-ci, le repère-image est tel que le plan formé par les deux axes  $x_{img}$ ,  $y_{img}$  est confondu avec le plan d'interprétation de l'un des segments du modèle 3D choisi comme segment référence. Là également, la transformation caméra-image est complètement indépendante du modèle. Le problème revient alors à calculer deux angles de rotation et la translation entre le repère-image et le repère-modèle. L'expression des deux contraintes d'alignement des vecteurs normaux des deux autres segments en fonction des angles de rotation fournit une équation polynomiale de degrés 8 et d'inconnue  $t = \tan(\phi)$  où  $\phi$  est l'un des deux angles de rotation cités précédemment. Cette équation est alors résolue par la méthode itérative de Bairstow. Connaissant la rotation finale, le problème de la translation devient un problème linéaire. En effet, en contraignant le vecteur position d'un point arbitraire de chaque segment à appartenir au plan d'interprétation du segment image correspondant, on obtient trois équations linéaires en fonction des composantes du vecteur translation recherché. Des simplifications dans les cas particuliers où les segments se coupent en un même point où sont coplanaires sont également présentées.

## Méthodes numériques

Ce sont des méthodes basées sur l'optimisation. Une fonction d'erreur est exprimée en fonction des paramètres de localisation. Cette fonction exprime généralement la somme des distances entre les primitives vues sur l'image et les estimations obtenues en projetant les primitives correspondantes sur le modèle. Certaines utilisent des approches classiques basées sur la dérivation telles que Levenberg-Marquardt ou Newton-Raphson, d'autres alternent



l'utilisation des modèles affine et perspectif de caméra en supposant la linéarité locale des équations de projection.

## Correspondances de points

Une première approche présentée par Lowe s'énonce comme suit ([Lowe87]) : Etant donné un ensemble de  $N$  points  $P_i(x_i, y_i, z_i)$  du modèle 3D et leur pixels projection respectifs  $(\hat{u}_i, \hat{v}_i)$  observés sur l'image. Grâce à une estimation initiale de la position et de l'orientation de la caméra, les points du modèles sont projetés sur l'image obtenant ainsi leurs projections théoriques  $(u_i, v_i)$ . La distance entre chaque point réellement vu sur l'image et l'estimation de sa projection est exprimée en fonction des six paramètres de rotation et de translation recherchés (et éventuellement de la distance focale de la caméra si elle est inconnue). Une fonction d'erreur est ainsi obtenue. La méthode consiste alors à corriger itérativement ces paramètres de localisation jusqu'à ce que cette fonction soit minimisée. En d'autres termes, on cherche à calculer  $\mathbf{R}$  et  $\mathbf{T}$  tels que la fonction suivante soit minimisée

$$E = \sum_{i=1}^{i=N} (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2 \quad (3.1.8)$$

avec

$$(u_i, v_i) = \left( f \cdot \frac{x_{ci}}{z_{ci}}, f \cdot \frac{y_{ci}}{z_{ci}} \right) \quad (3.1.9)$$

et

$$(x_{ci}, y_{ci}, z_{ci}) = \mathbf{R}(\mathbf{P}_i - \mathbf{T}) \quad (3.1.10)$$

Pour minimiser cette fonction, Lowe choisit d'adopter la méthode de Newton. Il s'agit donc de calculer les dérivées partielles des  $(u_i, v_i)$  en question par rapport à chacun des paramètres de position et d'orientation. Pour que les expressions des dérivées puissent être calculées de façon simple, l'auteur exprime la translation dans un repère-image plutôt que dans un repère classique. Les équations de projection deviennent alors

$$(x_{ci}, y_{ci}, z_{ci}) = \mathbf{R} \cdot \mathbf{P}_i \text{ et } (u_i, v_i) = \left( f \cdot \frac{x_{ci}}{z_{ci} + D_z} + D_x, f \cdot \frac{y_{ci}}{z_{ci} + D_z} + D_y \right) \quad (3.1.12)$$

$[D_x, D_y, D_z]$  est le nouveau vecteur de translation.  $D_z$  représente la distance du centre optique au modèle.  $D_x$  et  $D_y$  représentent la position du modèle dans le plan image. Cette nouvelle formulation comprend une approximation affine qui suppose que les mêmes translations  $D_x$  et  $D_y$  sont à appliquer aux différents points du modèle. Or, si l'on tient compte d'une projection perspective réelle les valeurs de  $D_x$  et  $D_y$  pour chaque point dépendent de la valeur  $z_{ci} + D_z$ .  $D_x$  et  $D_y$  sont donc les mêmes uniquement pour les points ayant une même valeur de  $z_{ci}$ . Lowe suppose également que les trois composantes de la correction à apporter à la rotation sont assez petites pour les considérer indépendantes les unes des autres.

L'approximation affine est une faiblesse de la méthode de Lowe. Elle a été améliorée dans [Ishii89] puis levée dans [Araujo96] qui présentent une nouvelle formulation de cette méthode dans laquelle les dérivées partielles sont calculées à partir des équations de projection perspective originale.

Une solution plus récente est présentée par Lu ([Lu00]). Son originalité est qu'elle prend en compte l'orthonormalité de la matrice de rotation. Il faut noter toutefois que l'optimisation de sa fonction d'erreur nécessite l'estimation des coordonnées des points de l'image dans le repère 3D de la caméra, ce qui revient en pratique à effectuer un appariement 3D-3D plutôt que 3D-2D. La méthode allie la rapidité et la convergence globale à l'inverse des méthodes d'optimisation classiques telles que Levenberg-Marquardt.

D'autres approches utilisent un modèle simplifié de caméra pour linéariser les équations obtenues dans un premier temps puis approcher itérativement la projection en pleine perspective. C'est le cas notamment pour Dementhon qui présente un algorithme en deux étapes utilisant quatre correspondances de points ou plus ([Dementhon95]). La première étape (POS: Pose from Orthography and Scaling) approxime la projection perspective par une projection orthographique et une mise à l'échelle. Il résout alors le système linéaire ce qui permet d'obtenir une estimation de la position et de l'orientation. La deuxième étape (POSIT : POS with Iterations) est itérative. Elle utilise à chaque pas les points estimés grâce aux résultats de POS plutôt que les points originaux pour que la projection orthographique se rapproche de plus en plus d'une projection perspective. L'avantage de cette approche par rapport aux méthodes de type Newton est de ne nécessiter aucune estimation initiale de la position et de l'orientation.

Horaud présente une façon plus simple d'obtenir le résultat de Dementhon et montre que la méthode appartient en fait à une classe de méthodes plus générale lorsque le modèle de caméra perspective est approximé à l'ordre zéro (perspective faible) ou à l'ordre un (para perspective) ([Horaud94]). Il démontre que la méthode utilisant le modèle para perspective a de meilleures propriétés de convergence et une meilleure robustesse en présence d'erreurs de calibrage. De plus, une façon simple de tenir compte de l'orthogonalité de la matrice de rotation est présentée.

Reprenant l'équation de projection d'un point en assumant, pour la clarté du raisonnement que la matrice des paramètres internes est  $\mathbf{K}=[\mathbf{I}_{3 \times 3} \mathbf{0}]$ . En posant  $\mathbf{I} = \frac{\mathbf{r}_1}{t_z}$ ,  $\mathbf{J} = \frac{\mathbf{r}_2}{t_z}$ ,  $x_0 = \frac{t_x}{t_z}$ ,

$y_0 = \frac{t_y}{t_z}$  on peut écrire

$$\begin{cases} u_i(1 + \varepsilon_i) - x_0 = \mathbf{I} \cdot \mathbf{P}_i \\ v_i(1 + \varepsilon_i) - y_0 = \mathbf{J} \cdot \mathbf{P}_i \end{cases} \quad (3.1.13)$$

En considérant un modèle à perspective faible ( $\varepsilon_i = \mathbf{P}_i \cdot \frac{\mathbf{r}_3}{t_z} = 0$ ), le système ( ) devient linéaire

en  $\mathbf{I}$  et  $\mathbf{J}$ . Il peut être résolu si l'on dispose d'au moins quatre correspondances de points. On retrouve alors la solution de Dementhon. Les équations de calcul de  $\mathbf{R}$  et  $\mathbf{t}$  à partir de  $\mathbf{I}$  et  $\mathbf{J}$  sont également données.

L'algorithme de calcul de la position et de l'orientation de la caméra est alors exécuté comme suit : à chaque itération, résoudre le système linéaire (6) et calculer  $\mathbf{R}$  et  $\mathbf{t}$ , mettre à jour les

$\varepsilon_i = \mathbf{P}_i \cdot \frac{\mathbf{r}_3}{t_z} = 0$ . L'algorithme s'arrête lorsque deux itérations successives donnent les mêmes valeurs de  $\varepsilon_i$ .

Pour le modèle para-perspective, la démarche est la même. Les expressions de  $\mathbf{I}$  et  $\mathbf{J}$  sont

toutefois remplacées par  $\mathbf{I}_p = \frac{\mathbf{r}_1 - x_0 \mathbf{r}_3}{t_z}$ ,  $\mathbf{J}_p = \frac{\mathbf{r}_2 - y_0 \mathbf{r}_3}{t_z}$ , et le système linéaire à résoudre est

$$\begin{cases} (u_i - x_0)(1 + \varepsilon_i) = \mathbf{I}_p \cdot \mathbf{P}_i \\ (v_i - y_0)(1 + \varepsilon_i) = \mathbf{J}_p \cdot \mathbf{P}_i \end{cases} \quad (3.1.14)$$

## Correspondances de droites

Lowe présente la façon d'étendre sa méthode aux correspondances de droites ([Lowe87]). Deux points arbitraires sont choisis sur chaque segment modèle apparié. De la même façon que pour les correspondances de points, les projections théoriques sont calculées en fonction des paramètres de position. On calcule ensuite les expressions des distances perpendiculaires  $d_i(\mathbf{R}, \mathbf{T})$  de chaque point théorique à la droite observée correspondante. La fonction d'erreur est alors

$$E = \sum_{i=1}^{i=2N} d_i^2 \quad (3.1.15)$$

où  $N$  est le nombre de segments appariés.

$d_i$  étant une expression linéaire des  $(u_i, v_i)$ , ses dérivées partielles par rapport aux paramètres de position et d'orientation sont également des combinaisons linéaires des dérivées partielles calculées dans le cas de correspondances de points. Un minimum de trois correspondances de droites permet alors de résoudre le problème. Cette méthode sera présentée plus en détails dans le chapitre suivant.

Liu ([Liu90]) reprend la technique des vecteurs normaux aux plans d'interprétations des segments. Il découple les problèmes de la rotation et de la translation. Pour calculer la rotation il reprend la contrainte d'alignement de Horaud exprimée par le produit scalaire entre le vecteur directeur de chaque segment 3D  $\mathbf{V}_i$  et le vecteur  $\mathbf{n}_i$  normal au plan d'interprétation du segment 2D correspondant :

$$\mathbf{n}_i \cdot (\mathbf{R}\mathbf{V}_i) = 0 \quad (3.1.16)$$

Deux façons de calculer la matrice de rotation sont alors possibles. La première solution est linéaire. Elle consiste à résoudre le système (3.1.16) en considérant les 9 composantes de la matrice comme indépendantes. La seconde est une solution itérative. La matrice  $\mathbf{R}$  est exprimée en fonction de ses 3 paramètres d'Euler  $\phi$ ,  $\theta$  et  $\psi$ . Le principe est que les angles de rotation sont obtenus en apportant de petites corrections  $\Delta\phi$ ,  $\Delta\theta$  et  $\Delta\psi$  à une valeur initiale  $\phi_0$ ,  $\theta_0$  et  $\psi_0$ . L'auteur utilise une expression  $L(\phi_0, \theta_0, \psi_0, \Delta\phi, \Delta\theta, \Delta\psi)$  qui est l'expression linéaire de l'équation (3.1.16) au voisinage de  $\phi_0$ ,  $\theta_0$ ,  $\psi_0$ .  $L$  est linéaire en fonction de  $\Delta\phi$ ,  $\Delta\theta$ ,  $\Delta\psi$ . Trois correspondances permettent de le résoudre. L'algorithme peut s'énoncer comme suit :

- 1) initialiser  $k=0$ ,  $\phi_k=0$ ,  $\theta_k=0$ ,  $\psi_k=0$  ;
- 2) résoudre  $L(\phi_k, \theta_k, \psi_k, \Delta\phi_k, \Delta\theta_k, \Delta\psi_k)=0$  ;
- 3) si  $|\Delta\phi_k|+|\Delta\theta_k|+|\Delta\psi_k|$  est inférieur à un seuil alors aller à 5) ;
- 4) sinon faire  $\phi_{k+1}=\phi_k+\Delta\phi_k$ ,  $\theta_{k+1}=\theta_k+\Delta\theta_k$ ,  $\psi_{k+1}=\psi_k+\Delta\psi_k$  ; faire  $k=k+1$  ; aller à 2) ;
- 5) fin.

Selon les auteurs, la méthode donne de bon résultats à condition que les trois angles d'Euler soient inférieurs à  $30^\circ$ .

Une fois la rotation connue, le calcul de la translation se fait grâce à la résolution du système d'équations suivant obtenu avec la contrainte d'alignement du vecteur position d'un point arbitraire  $\mathbf{P}_i$  de chaque segment 3D apparié avec le plan d'interprétation correspondant

$$\mathbf{n}_i \cdot (\mathbf{R}\mathbf{P}_i + \mathbf{t}) = 0 \quad (3.1.17)$$

Ce système est linéaire par rapport à  $\mathbf{t}$ . Il est à noter que le découplage de la rotation et de la translation, s'il réduit le coût en temps de calcul et les non linéarités, donne toutefois une précision moindre ([Horaud93]).

Phong et Horaud utilisent les contraintes d'alignement (7) et (8) pour en déduire une fonction d'erreur ([Phong95]).

$$f(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^N (\mathbf{n}_i \cdot (\mathbf{R}\mathbf{V}_i))^2 + (\mathbf{n}_i \cdot (\mathbf{R}\mathbf{P}_i + \mathbf{t}))^2 \quad (3.1.18)$$

L'originalité de leur approche est l'introduction des quaternions pour représenter la rotation. La fonction d'erreur a alors une forme quadratique. Elle est minimisée grâce à un algorithme dit «région de confiance» permettant de s'affranchir du problème de l'initialisation. Cette méthode sera présentée plus en détail dans le chapitre suivant.

Christy présente, comme pour le cas des correspondances de points, une méthode qui utilise l'estimation de la projection par des modèles à perspective faible ou para perspective pour converger à terme vers une solution proche de la solution en perspective réelle ([Christy98]). Reprenons les mêmes notations que pour la méthode de Horaud. Soit un point  $\mathbf{P}_i$  appartenant à un segment 3D de vecteur directeur  $\mathbf{D}_i$ . On peut écrire

$$\mathbf{P}_i = \mathbf{Q}_i + \lambda \mathbf{D}_i$$

L'équation de sa projection sur l'image s'écrit :

$$\begin{pmatrix} su_i \\ s \end{pmatrix} = \mathbf{M} \begin{pmatrix} \mathbf{P}_i \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{I} \cdot \mathbf{Q}_i + x_0 \\ \mathbf{J} \cdot \mathbf{Q}_i + y_0 \\ 1 + \eta_i \end{pmatrix} + \lambda \begin{pmatrix} \mathbf{I} \cdot \mathbf{D}_i \\ \mathbf{J} \cdot \mathbf{D}_i \\ \mu_i \end{pmatrix} \quad (3.1.19)$$

avec  $\eta_i = \mathbf{K} \cdot \mathbf{Q}_i$  et  $\mu_i = \mathbf{K} \cdot \mathbf{D}_i$ .

D'autre part, le point projeté est contraint d'appartenir à une droite image  $a_i u + b_i v + c_i = 0$ .

En remplaçant les termes de (10) dans cette dernière équation et compte tenu du fait que ce résultat doit être vérifié pour tout point du segment ( $\lambda$  quelconque), on obtient les deux équations suivantes pour chaque segment apparié à une droite image

$$\begin{cases} a_i \mathbf{I} \cdot \mathbf{Q}_i + b_i \mathbf{J} \cdot \mathbf{Q}_i + a_i x_0 + b_i y_0 + c_i (1 + \eta_i) = 0 \\ a_i \mathbf{I} \cdot \mathbf{D}_i + b_i \mathbf{J} \cdot \mathbf{D}_i + c_i \mu_i = 0 \end{cases} \quad (3.1.20)$$

Les auteurs dérivent ensuite les expressions du système (11) dans le cas d'un modèle à faible perspective :

$$\begin{cases} a_i \mathbf{I} \cdot \mathbf{Q}_i + b_i \mathbf{J} \cdot \mathbf{Q}_i + a_i x_0 + b_i y_0 + c_i = 0 \\ a_i \mathbf{I} \cdot \mathbf{D}_i + b_i \mathbf{J} \cdot \mathbf{D}_i = 0 \end{cases} \quad (3.1.21)$$

Puis dans le cas d'un modèle para-perspective :

$$\begin{cases} a_i \mathbf{I}_p \cdot \mathbf{Q}_i + b_i \mathbf{J}_p \cdot \mathbf{Q}_i + a_i x_0 + b_i y_0 + c_i = 0 \\ a_i \mathbf{I}_p \cdot \mathbf{D}_i + b_i \mathbf{J}_p \cdot \mathbf{D}_i = 0 \end{cases} \quad (3.1.22)$$

Le système (12) est linéaire par rapport à  $\mathbf{I}$ ,  $\mathbf{J}$ ,  $x_0$ ,  $y_0$  et peut être résolu avec au moins quatre correspondances de droite. Le système (13) est linéaire par rapport à  $\mathbf{I}_p$ ,  $\mathbf{J}_p$ ,  $x_0$ ,  $y_0$  et peut également être résolu avec quatre correspondances. La convergence itérative vers les paramètres de position et d'orientation du modèle perspectif se fait de la même façon que pour les correspondances de points,  $\eta_i$  et  $\mu_i$  jouant ici le même rôle que  $\varepsilon_i$ .

L'auteur présente également une analyse des topologies particulières des segments utilisés pour les correspondances 3D-2D qui représentent des situations singulières. Il s'agit notamment du cas de segments parallèles ou de segments se coupant en un même point.

## Utilisation simultanée de points et de droites

Les méthodes mentionnées ci-dessus exploitent des critères basés sur un même type de correspondances (droite-droite ou point-point). Il est possible d'exprimer un critère hétérogène utilisant à la fois les deux types de correspondances. Ce critère est difficile à exploiter car il mélange des termes de dimensions différentes (écarts d'orientation exprimés en degrés et écarts de distance exprimés en mètres). Il est alors nécessaire de pondérer ces termes différemment et le calcul des paramètres de pondération est souvent délicat. Borges ([Borges00]) propose une méthode exploitant à la fois des correspondances de droites et des correspondances de points grâce à un changement de représentation de ces primitives (*représentation unifiée*). Il obtient ainsi un critère homogène. Un estimateur optimal de type Moindres Carrés Pondérés (MCP) est également proposé ([Borges02]). Cet algorithme minimise le critère de façon optimale et non itérative.

## Discussion

A la lumière de cet état de l'art, nous allons mettre en parallèle les caractéristiques du contexte de la localisation en robotique mobile d'une part et les caractéristiques des méthodes ci-dessus d'autre part. Le but est de choisir, voire d'adapter, les techniques qui répondent le mieux aux exigences de cette tâche.

### Type de primitives de correspondances

L'environnement de notre application est de type appartement. Il est composé essentiellement d'un ensemble de plans (murs, plafond, sol,...) et de formes plus ou moins polyédriques (gros meubles, portes, fenêtres,...). Une segmentation en contours d'une vue de cet environnement mettrait donc en évidence les arêtes formées par les intersections de ces plans. Ces arêtes sont des droites. De plus l'extraction et l'approximation polygonale de ce type de primitives est peu sensible à la présence de bruit et les paramètres de droite calculés ne sont que faiblement modifiés par les occultations partielles. Ceci nous oriente plutôt vers des correspondances de droites.

### Technique de calcul

Comme dans toute application pratique, il existe plusieurs sources de bruit, que ce soit au niveau de l'acquisition de l'image elle-même, de la segmentation ou encore du calibrage (paramètres intrinsèques). Les contraintes dérivées de ces mesures produisent donc des équations avec des paramètres non exacts. L'utilisation de techniques d'optimisation numérique est dans ce cas plus indiquée. De plus, le système dispose dans la majorité des cas d'une estimation de la position et de l'orientation grâce à l'odométrie, ce qui est d'une grande importance si l'on adopte une solution non linéaire. La sensibilité par rapport à la qualité de la solution initiale devra toutefois être étudiée.

### Modèle de caméra

Les dimensions des primitives que le système est amené à utiliser sont du même ordre de grandeur (si ce n'est plus grandes) que leur distance au centre optique. Dans ce cas, les effets de perspective sont très accentués. Une petite variation d'orientation peut aboutir à des variations très importantes sur les primitives de l'image. La méthode utilisée devra donc impérativement être basée sur un modèle de caméra en pleine perspective. Ceci a pour effet de produire des équations fortement non linéaires et donc d'augmenter les temps de convergence lors de l'optimisation numérique.

## **Temps de calculs**

Le système de localisation doit pouvoir être sollicité en cours de mouvement du robot (localisation en ligne). Le processus doit donc avoir des caractéristiques temps-réel (ou proche du temps réel). La méthode retenue doit donc réduire le coût de calcul et les temps de convergence (nombre d'itérations). Une façon de faire est d'une part de réduire le nombre d'inconnues à optimiser, d'autre part de rechercher quand cela est possible des solutions proches de la linéarité.

## **Configurations singulières**

Les méthodes précédentes présentent des singularités lorsque les segments du modèle sont disposés selon certaines configurations. Ces configurations sont présentées dans [Christy98] . Il s'agit notamment du cas où tous les segments se coupent en un même point et du cas de segments parallèles. Ces configurations sont très fréquentes dans des environnement intérieurs (coins, arêtes de portes,...). Ces singularités devront donc être prises en compte lors du développement de la méthode de localisation de notre robot.

## Méthodes développées

Dans la partie bibliographique, nous avons passé en revue les différentes techniques existantes pour recouvrir la position et l'orientation d'une caméra à partir d'une seule image et étant donné un ensemble de correspondances entre des primitives géométriques 2D de cette image et des éléments géométriques 3D d'un modèle de la scène. Nous avons mis en évidence les différences de chaque approche quant à l'approche mathématique de résolution, au formalisme de description des paramètres de position et au type de correspondances utilisées. L'analyse du contexte de la robotique mobile nous a amené aux conclusions que la méthode de calcul de la position et de l'orientation utilisée devra exploiter des techniques d'optimisation numériques, utiliser des correspondances de droites et se baser sur un modèle perspectif de caméra. L'efficacité et la robustesse par rapport à la qualité d'initialisation devront également être optimisées.

Dans la partie qui va suivre nous présentons deux méthodes numériques différentes de calcul de position et d'orientation de la caméra. Elles sont basées sur un modèle perspectif de caméra et utilisent des correspondances de droites. La première est basée sur l'approche de Lowe et la seconde sur l'approche de Phong-Horaud. Nous présentons ensuite leur adaptation au contexte de la robotique mobile pour mieux satisfaire aux exigences énoncées plus haut. Une évaluation comparative des méthodes sur des données synthétiques et sur des images réelles termine le chapitre.

## Méthode basée sur l'approche de Lowe

### Formulation d'origine

Soit un repère 3D d'origine  $O$  lié au modèle 3D de l'environnement dans lequel se meut un robot mobile. Soit un repère d'origine  $O_c$  lié à la caméra tel que  $O_c$  soit confondu avec le centre optique et l'axe des  $z$  confondu avec le rayon optique (figure 2.1). Soit un point  $P(x,y,z)$  du modèle 3D exprimé dans le repère lié à ce dernier. Les coordonnées  $(u,v)$  du pixel de projection de  $P$  sur l'image s'obtient de la façon suivante :

$$(x_c, y_c, z_c) = \mathbf{R} \cdot (\mathbf{P} - \mathbf{T}) \quad (3.2.1)$$

$$(u, v) = \left( f \cdot \frac{x_c}{z_c}, f \cdot \frac{y_c}{z_c} \right) \quad (3.2.2)$$

où  $\mathbf{R}$  et  $\mathbf{T}$  sont la matrice de rotation et le vecteur de translation qui relie le repère lié à la caméra et le repère du modèle et  $f$  la distance focale.

Le problème est le suivant : étant donné un ensemble de  $N$  points  $P_i (i=1, \dots, N)$  du modèle 3D de projections théoriques respectives  $(u_i, v_i)$  et l'ensemble des pixels correspondants  $(\hat{u}_i, \hat{v}_i)$  observés sur l'image, calculer  $\mathbf{R}$  et  $\mathbf{T}$  minimisant la fonction d'erreur suivante

$$E = \sum_{i=1}^{i=N} (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2 \quad (3.2.3)$$

C'est un problème de moindres carrés non linéaires. En vue d'utiliser la méthode de Newton, il nous faut d'abord calculer les expressions des dérivées partielles de  $u$  et  $v$  en fonction de chacun des six paramètres qui forment  $\mathbf{R}$  et  $\mathbf{T}$ . Il est difficile de calculer ces dérivées à partir de cette formulation. Pour rendre plus aisé le calcul des dérivées, Lowe a d'abord reformulé la translation :

$$(x_c, y_c, z_c) = \mathbf{R} \cdot \mathbf{P} \quad (3.2.4)$$

$$(u, v) = (f \cdot \frac{x_c}{z_c + D_z} + D_x, f \cdot \frac{y_c}{z_c + D_z} + D_y) \quad (3.2.5)$$

Dans cette formulation  $\mathbf{R}$  garde la même signification que précédemment tandis que, sous une approximation affine,  $D_x, D_y$  spécifient la position du point sur le plan image et  $D_z$  la distance entre les points et le centre optique de la caméra.

Reste à résoudre le problème de l'expression de la rotation. Lowe note que pour l'application de la méthode de Newton il n'y a besoin que de la possibilité de modifier itérativement les valeur de la première estimation de  $\mathbf{R}$ . La méthode adoptée consiste à calculer la correction de la rotation initiale par une suite de trois rotations dans des directions mutuellement orthogonales  $\Psi, \theta, \phi$  (respectivement autour des axes  $x_c, y_c$  et  $z_c$  du repère caméra) et d'exprimer les dérivées partielles en fonctions de ces rotations. Les corrections sont supposées indépendantes puisque de faible dimension. Lowe obtient ainsi le tableau des dérivées partielles de  $u$  et  $v$  en fonction des six paramètres de rotation et de translation considérés :

	$u$	$v$
$D_x$	1	0
$D_y$	0	1
$D_z$	$-fc^2x_c$	$-fc^2y_c$
$\Psi$	$-fc^2x_cy_c$	$-fc(z_c+y_c^2)$
$\theta$	$-fc(z_c+x_c^2)$	$fc^2x_c$
$\phi$	$cx_c$	$cy_c$

**Table 3.1** : dérivées partielles de  $u$  et  $v$  par rapport aux paramètres de position et d'orientation

avec  $c = \frac{1}{z_c + D_z}$

A chaque itération il s'agit de calculer le vecteur de correction  $[\Delta D_x, \Delta D_y, \Delta D_z, \Delta \Psi, \Delta \theta, \Delta \phi]$  solution du système linéaire suivant

$$\begin{cases} \frac{\partial u}{\partial D_x} \Delta D_x + \frac{\partial u}{\partial D_y} \Delta D_y + \frac{\partial u}{\partial D_z} \Delta D_z + \frac{\partial u}{\partial \Psi} \Delta \Psi + \frac{\partial u}{\partial \theta} \Delta \theta + \frac{\partial u}{\partial \phi} \Delta \phi = E_u \\ \frac{\partial v}{\partial D_x} \Delta D_x + \frac{\partial v}{\partial D_y} \Delta D_y + \frac{\partial v}{\partial D_z} \Delta D_z + \frac{\partial v}{\partial \Psi} \Delta \Psi + \frac{\partial v}{\partial \theta} \Delta \theta + \frac{\partial v}{\partial \phi} \Delta \phi = E_v \end{cases}$$

puis de mettre à jour  $\mathbf{R}$  et  $\mathbf{T}$  et de calculer la nouvelle valeur de  $E$ . La mise à jour de  $\mathbf{R}$  se fait en constituant une matrice de rotation  $d\mathbf{R}$  composée des trois corrections des rotations élémentaires obtenues et en faisant le produit avec l'ancienne matrice  $\mathbf{R}$ . Le vecteur translation est, quant à lui, mis à jour par l'équation suivante :

$$\mathbf{T} = \mathbf{R}^{-1} \left[ -\frac{D_x(z_c + D_z)}{f}, -\frac{D_y(z_c + D_z)}{f}, -D_z \right]^T \quad (3.2.6)$$

Le processus est répété jusqu'à ce qu'il y ait convergence (quand la valeur de l'erreur n'évolue plus de façon significative). Chaque correspondance de points fournissant deux équations, il nous faut donc au moins trois points pour pouvoir utiliser la méthode.

### Variante de Araujo-Carceroni

Une des faiblesses de l'approche précédente est l'approximation affine selon laquelle Lowe considèrent que  $D_x$  et  $D_y$  ont la même valeur pour tous les points considérés. Ceci n'est pas



forcément le cas en pratique puisque si l'on applique les lois de la projection perspective  $D_x$  et  $D_y$  dépendent de la distance de chaque point au centre optique. Cette approximation à été levée dans [Araujo96] en respectant dans les expressions de  $u$  et  $v$  la projection perspective :

$$(u, v) = \left( f \cdot \frac{x_c + D_x}{z_c + D_z}, f \cdot \frac{y_c + D_y}{z_c + D_z} \right) \quad (3.2.7)$$

Dans cette formulation le vecteur  $T$  se calcule toujours grâce à l'équation (3.2.6). Les nouvelles expressions des dérivées partielles sont données dans le tableau ci-dessous. La suite de la méthode reste identique au principe utilisé par Lowe.

	$u_i$	$V_i$
$D_x$	$\alpha_u c$	0
$D_y$	0	$\alpha_v c$
$D_z$	$-\alpha_u a c^2$	$-\alpha_v b c^2$
$\Psi$	$-\alpha_u a c^2 y_c$	$-\alpha_v c(z_c + b c y_c)$
$\theta$	$\alpha_u c(z_c + a c x_c)$	$\alpha_v b c^2 x_c$
$\phi$	$-\alpha_u c y_c$	$\alpha_v c x_c$

**Table 3.2** : dérivées partielles de  $u$  et  $v$  par Carceroni

avec  $[a \ b \ c] = [x_c + D_x \ y_c + D_y \ 1/z_c + D_z]$ .

## Passage aux correspondances de droites

Une droite observée sur l'image est caractérisée par sa pente  $\hat{\rho}_i$  et la distance perpendiculaire à l'origine du repère  $\hat{d}_i$  liées par l'équation suivante :

$$\hat{d}_i = \cos(\hat{\rho}_i) \cdot \hat{u}_i + \sin(\hat{\rho}_i) \cdot \hat{v}_i \quad (3.2.8)$$

dans laquelle  $(\hat{u}_i, \hat{v}_i)$  est un point appartenant à cette droite. Pour un autre point quelconque  $(u, v)$  la distance à l'origine dans la même direction est

$$d_i = \cos(\hat{\rho}_i) \cdot u_i + \sin(\hat{\rho}_i) \cdot v_i \quad (3.2.9)$$

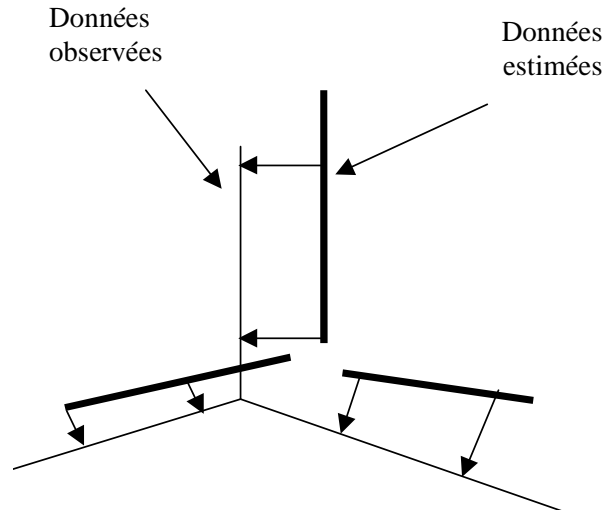
La distance entre le point  $(u_i, v_i)$  et la droite considérée est donc tout simplement  $\hat{d}_i - d_i$ . Si  $(u_i, v_i)$  est la projection théorique d'un point 3D  $P_i(x_i, y_i, z_i)$  appartenant à un segment du modèle, la distance théorique  $d_i$  peut alors être exprimée en fonction des six paramètres de position et d'orientation en remplaçant dans (3.2.9)  $u_i$  et  $v_i$  par leurs expressions issues de (3.2.7). En prenant deux points arbitraire de chaque segment, la fonction d'erreur, dans le cas de correspondances de droite, peut alors être écrite comme suit

$$E = \sum_{i=1}^{i=N} (d_{1i} - \hat{d}_i)^2 + (d_{2i} - \hat{d}_i)^2 \quad (3.2.10)$$

ou encore

$$E = \sum_{i=1}^{i=N} (\cos(\hat{\rho}_i) \cdot u_{1i} + \sin(\hat{\rho}_i) \cdot v_{1i} - \hat{d}_i)^2 + (\cos(\hat{\rho}_i) \cdot u_{2i} + \sin(\hat{\rho}_i) \cdot v_{2i} - \hat{d}_i)^2 \quad (3.2.11)$$

Chaque correspondance entre un segment du modèle et une droite de l'image fournit deux équations. Il faut donc un minimum de 3 correspondances pour recouvrir les six inconnues.



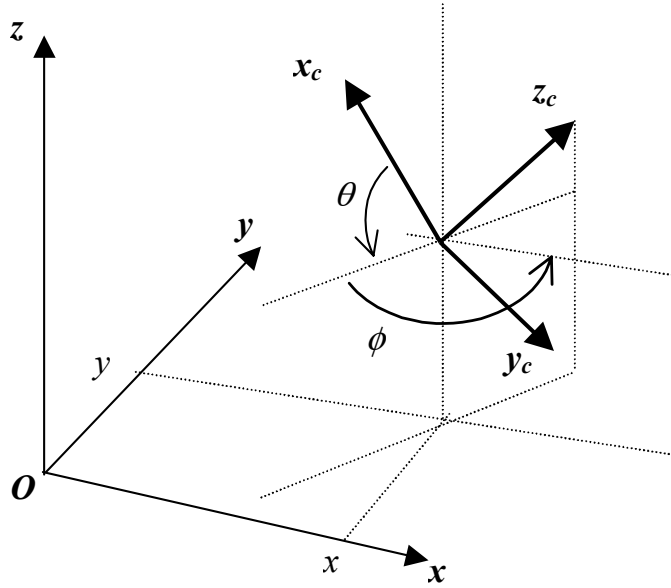
*Figure 3.5 : utilisation des appariements de segments de droites dans la méthode de Lowe*

### Adaptation au contexte de la robotique mobile

Nous allons à présent formuler la méthode présentée de façon à ce qu'elle soit plus adaptée au contexte de la localisation en robotique mobile. Même si le robot opère dans un environnement 3D, le déplacement du repère caméra est en réalité défini dans un plan horizontal situé à une hauteur fixe. Cette contrainte permet de réduire le nombre de degrés de liberté du système à trois, un pour la rotation et deux pour la translation. La rotation entre les deux repères est constituée d'une rotation connue  $\theta$  autour de l'axe  $y$  du repère caméra et d'une rotation inconnue  $\phi$  autour du nouvel axe  $z$  obtenu après la première rotation (figure 3.6).  $\mathbf{R}$  s'exprime alors comme suit

$$\mathbf{R} = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta)\cos(\phi) & -\cos(\theta)\sin(\phi) & \sin(\theta) \\ \sin(\phi) & \cos(\phi) & 0 \\ -\sin(\theta)\cos(\phi) & \sin(\theta)\sin(\phi) & \cos(\theta) \end{pmatrix} \quad (3.2.12)$$

Le vecteur translation quant à lui, s'il est exprimé dans le repère objet, s'écrit  $\mathbf{T}=[x,y,z]^T$  avec  $z$  constant et connu. Le problème consiste donc à calculer trois variables  $\phi$ ,  $x$  et  $y$ .



**Figure 3.6 :** degrés de liberté de la caméra en robotique mobile

Avec la réduction du nombre de degrés de liberté, l'approche de Lowe peut être formulée en exprimant  $\mathbf{R}$  en fonction de ces paramètres que sont les rotations élémentaires autour des axes principaux. Pour chaque point  $\mathbf{P}_i(x_i, y_i, z_i)$  l'équation de projection s'écrit donc :

$$\begin{cases} u_i = f \cdot \frac{(x_i - x) \cdot \cos(\theta) \cdot \cos(\phi) - (y_i - y) \cdot \cos(\theta) \cdot \sin(\phi) + (z_i - z) \cdot \sin(\theta)}{- (x_i - x) \cdot \sin(\theta) \cdot \cos(\phi) + (y_i - y) \cdot \sin(\theta) \cdot \sin(\phi) + (z_i - z) \cdot \cos(\theta)} \\ v_i = f \cdot \frac{(x_i - x) \cdot \sin(\phi) + (y_i - y) \cdot \cos(\phi)}{- (x_i - x) \cdot \sin(\theta) \cdot \cos(\phi) + (y_i - y) \cdot \sin(\theta) \cdot \sin(\phi) + (z_i - z) \cdot \cos(\theta)} \end{cases} \quad (3.2.13)$$

Un avantage de cette formulation est que le vecteur translation est exprimé dans le repère modèle. Ceci permet de fixer  $z$  à une valeur fixe et connue qui est la hauteur de la caméra embarquée sur le robot. Il est également à noter que la réduction du nombre de variables permet de réduire le nombre minimal de primitives nécessaires à la résolution des équations. Cependant, cette écriture reste complexe en raison du fort couplage entre la rotation et la translation représenté par les termes produits  $x \cos(\theta)$ ,  $y \cos(\theta)$ ,  $x \sin(\theta)$ ,  $y \sin(\theta)$ . Si l'on exprime la translation dans le repère caméra plutôt que dans le repère modèle, l'équation (1) devient

$$(x_c, y_c, z_c) = \mathbf{R}\mathbf{P} + \mathbf{t} \quad (3.2.14)$$

ce qui donne les équations de projection suivantes :

$$\begin{cases} u_i = f \cdot \frac{x_i \cdot \cos(\theta) \cdot \cos(\phi) - y_i \cdot \cos(\theta) \cdot \sin(\phi) + z_i \cdot \sin(\theta) + x_c}{- x_i \cdot \sin(\theta) \cdot \cos(\phi) + y_i \cdot \sin(\theta) \cdot \sin(\phi) + z_i \cos(\theta) + z_c} \\ v_i = f \cdot \frac{x_i \cdot \sin(\phi) + y_i \cdot \cos(\phi) + y_c}{- x_i \cdot \sin(\theta) \cdot \cos(\phi) + y_i \cdot \sin(\theta) \cdot \sin(\phi) + z_i \cos(\theta) + z_c} \end{cases} \quad (3.2.15)$$

La translation et la rotation sont maintenant mieux découplées mais le nombre de degrés de liberté est apparemment plus élevé. On a donc à résoudre une fonction à quatre inconnues alors que le système physique est à 3 degrés de liberté. En fait, compte tenu de la configuration décrite plus haut, la composante  $x_c$  n'est pas indépendante de  $z_c$ . Elle peut être exprimée en fonction de cette dernière :

$$\mathbf{t} = -\mathbf{R}\mathbf{T} \quad (3.2.16)$$

d'où

$$\begin{cases} x_c = -x \cdot \cos(\theta) \cdot \cos(\phi) + y \cdot \cos(\theta) \cdot \sin(\phi) - z \cdot \sin(\theta) \\ z_c = x \cdot \sin(\theta) \cdot \cos(\phi) - y \cdot \sin(\theta) \cdot \sin(\phi) - z \cdot \cos(\theta) \end{cases} \quad (3.2.17)$$

L'expression obtenue est alors :

$$x_c = -\cot(\theta) \cdot z_c - \frac{z}{\sin(\theta)}$$

On revient donc à un système à 3 degrés de liberté d'inconnues  $\phi, y_c, z_c$ . On peut à présent réécrire les expressions de  $u$  et  $v$  sous la forme suivante :

$$\begin{cases} u_i = \alpha_u \cdot \frac{x_i \cdot \cos(\theta) \cdot \cos(\phi) - y_i \cdot \cos(\theta) \cdot \sin(\phi) + z_i \cdot \sin(\theta) - \cot(\theta) \cdot z_c - z / \sin(\theta)}{-x_i \cdot \sin(\theta) \cdot \cos(\phi) + y_i \cdot \sin(\theta) \cdot \sin(\phi) + z_i \cos(\theta) + z_c} + u_0 \\ v_i = \alpha_v \cdot \frac{x_i \cdot \sin(\phi) + y_i \cdot \cos(\phi) + y_c}{-x_i \cdot \sin(\theta) \cdot \cos(\phi) + y_i \cdot \sin(\theta) \cdot \sin(\phi) + z_i \cos(\theta) + z_c} + v_0 \end{cases} \quad (3.2.18)$$

L'équation (3.2.9) peut être réécrite comme suit :

$$d_i = \frac{\alpha_u \cdot \cos(\hat{\rho}) \cdot (x_i \cdot \cos(\theta) \cdot \cos(\phi) - y_i \cdot \cos(\theta) \cdot \sin(\phi) + z_i \cdot \sin(\theta) - \cot(\theta) \cdot z_c - z / \sin(\theta)) + \alpha_v \cdot \sin(\hat{\rho}) \cdot (x_i \cdot \sin(\phi) + y_i \cdot \cos(\phi) + y_c)}{-x_i \cdot \sin(\theta) \cdot \cos(\phi) + y_i \cdot \sin(\theta) \cdot \sin(\phi) + z_i \cos(\theta) + z_c} + d_{0i} \quad (3.2.19)$$

Si l'on considère sur chaque segment  $L_i$  du modèle 3D deux points  $P_{1i}(x_{1i}, y_{1i}, z_{1i})$  et  $P_{2i}(x_{2i}, y_{2i}, z_{2i})$  la fonction d'erreur (3.2.10) devient

$$E = \sum_{i=1}^{i=N} \left( \frac{A_{1i} \cdot \sin(\phi) + B_{1i} \cdot \cos(\phi) + C_{1i} y_c + D_{1i} z_c + E_{1i}}{F_{1i} \cdot \cos(\phi) + G_{1i} \cdot \sin(\phi) + H_{1i} z_c + I_{1i}} - \hat{d}_i \right)^2 + \left( \frac{A_{2i} \cdot \sin(\phi) + B_{2i} \cdot \cos(\phi) + C_{2i} y_c + D_{2i} z_c + E_{2i}}{F_{2i} \cdot \cos(\phi) + G_{2i} \cdot \sin(\phi) + H_{2i} z_c + I_{2i}} - \hat{d}_i \right)^2 \quad (3.2.20)$$

avec :

$$\begin{aligned} A_{ij} &= \alpha_u \cdot x_{ij} \cdot \cos(\hat{\rho}_i) \cdot \cos(\theta) + \alpha_v \cdot y_{ij} \cdot \cos(\hat{\rho}_i) \\ B_{ij} &= -\alpha_u \cdot y_{ij} \cdot \cos(\hat{\rho}_i) \cdot \cos(\theta) + \alpha_v \cdot x_{ij} \cdot \sin(\hat{\rho}_i) \\ C_{ij} &= \alpha_v \cdot \sin(\hat{\rho}_i) \\ D_{ij} &= \alpha_u \cdot \cos(\hat{\rho}_i) \cdot \cot(\theta) \\ E_{ij} &= \alpha_u \cdot \cos(\hat{\rho}_i) \cdot (z_{ij} \cdot \sin(\theta) - \frac{z}{\sin(\theta)}) \\ F_{ij} &= -x_{ij} \cdot \sin(\theta) \\ G_{ij} &= y_{ij} \cdot \sin(\theta) \\ H_{ij} &= 1 \\ I_{ij} &= z_{ij} \cdot \cos(\theta) \end{aligned}$$

pour  $i=1,2$  et  $j=1,2,\dots,N$ .

En posant comme vecteur-inconnue  $\mathbf{q}=[\cos(\phi) \sin(\phi) y_c z_c]$  la fonction d'erreur devient

$$E = \sum_{i=1}^{i=N} \left( \frac{A_{1i} \cdot q_1 + B_{1i} \cdot q_2 + C_{1i} q_3 + D_{1i} \cdot q_4 + E_{1i}}{F_{1i} \cdot q_1 + G_{1i} \cdot q_2 + H_{1i} q_3 + I_{1i}} - \hat{d}_i \right)^2 + \left( \frac{A_{2i} \cdot q_1 + B_{2i} \cdot q_2 + C_{2i} q_3 + D_{2i} \cdot q_4 + E_{2i}}{F_{2i} \cdot q_1 + G_{2i} \cdot q_2 + H_{2i} q_3 + I_{2i}} - \hat{d}_i \right)^2 + \lambda \cdot (q_1^2 + q_2^2 - 1)^2 \quad (3.2.21)$$

Le terme  $\lambda \cdot (q_1^2 + q_2^2 - 1)^2$  a été ajouté pour garantir l'orthogonalité de la matrice de rotation obtenue. Le scalaire  $\lambda$  permet de pondérer l'influence de cette contrainte.

Les dérivées partielles de  $d$  par rapport aux paramètres  $q_i$  s'expriment donc comme suit :

$$\begin{aligned} \frac{\partial d_{ij}}{\partial q_1} &= \frac{(A_{ij} \cdot G_{ij} - F_{ij} \cdot B_{ij}) \cdot q_2 + (A_{ij} \cdot H_{ij} - F_{ij} \cdot C_{ij}) q_3 - F_{ij} D_{ij} \cdot q_4 + A_{ij} \cdot I_{ij} - F_{ij} \cdot E_{ij}}{(F_{ij} \cdot q_1 + G_{ij} \cdot q_2 + H_{ij} q_3 + I_{ij})^2} \\ \frac{\partial d_{ij}}{\partial q_2} &= \frac{(A_{ij} \cdot G_{ij} - F_{ij} \cdot B_{ij}) \cdot q_2 + (A_{ij} \cdot H_{ij} - F_{ij} \cdot C_{ij}) q_3 - F_{ij} D_{ij} \cdot q_4 + A_{ij} \cdot I_{ij} - F_{ij} \cdot E_{ij}}{(F_{ij} \cdot q_1 + G_{ij} \cdot q_2 + H_{ij} q_3 + I_{ij})^2} \\ \frac{\partial d_{ij}}{\partial q_3} &= \frac{(A_{ij} \cdot G_{ij} - F_{ij} \cdot B_{ij}) \cdot q_2 + (A_{ij} \cdot H_{ij} - F_{ij} \cdot C_{ij}) q_3 - F_{ij} D_{ij} \cdot q_4 + A_{ij} \cdot I_{ij} - F_{ij} \cdot E_{ij}}{(F_{ij} \cdot q_1 + G_{ij} \cdot q_2 + H_{ij} q_3 + I_{ij})^2} \\ \frac{\partial d_{ij}}{\partial q_4} &= \frac{(A_{ij} \cdot G_{ij} - F_{ij} \cdot B_{ij}) \cdot q_2 + (A_{ij} \cdot H_{ij} - F_{ij} \cdot C_{ij}) q_3 - F_{ij} D_{ij} \cdot q_4 + A_{ij} \cdot I_{ij} - F_{ij} \cdot E_{ij}}{(F_{ij} \cdot q_1 + G_{ij} \cdot q_2 + H_{ij} q_3 + I_{ij})^2} \end{aligned}$$

pour  $i=1,2$  et  $j=1,2,\dots,N$ .

La méthode de Gauss-Newton ou de Levenberg-Marquardt peut à présent être utilisée pour minimiser la fonction d'erreur.

## Méthode basée sur l'approche de Phong-Horaud

### Formulation d'origine

L'approche précédente a pour principe la minimisation de la distance entre chaque point vu sur l'image et la projection estimée du point lui correspondant sur le modèle. Phong et Horaud adoptent une approche différente dans laquelle la différence entre ce qui est vu sur l'image et ce qui est estimé à partir du modèle s'exprime par des différences d'orientations de vecteurs dans l'espace 3D, différence qui se traduit par la valeur de leur produit scalaire ([Phong95]). Soit  $L_i$  un segment de droite 3D appartenant au modèle et dont  $\mathbf{V}$  et  $\mathbf{P}$  sont respectivement les vecteurs directeur et de position ( $\mathbf{P}$  représente les coordonnées d'un point arbitraire  $\mathbf{M}(x,y,z)$  dans le repère modèle).  $\mathbf{V}_c = \mathbf{R}\mathbf{V}$  et  $\mathbf{P}_c = \mathbf{R}\mathbf{P} + \mathbf{t}$  sont les transformées respectives de  $\mathbf{V}$  et  $\mathbf{P}$  dans le repère caméra. Soit  $l$  le vecteur vu sur l'image comme étant la projection de  $L$ .

Le point  $\mathbf{m}(u,v)$  de l'image, projection de  $\mathbf{M}$  appartient à la droite image  $l$  d'équation de la forme  $a \cdot u + b \cdot v + c = 0$ .

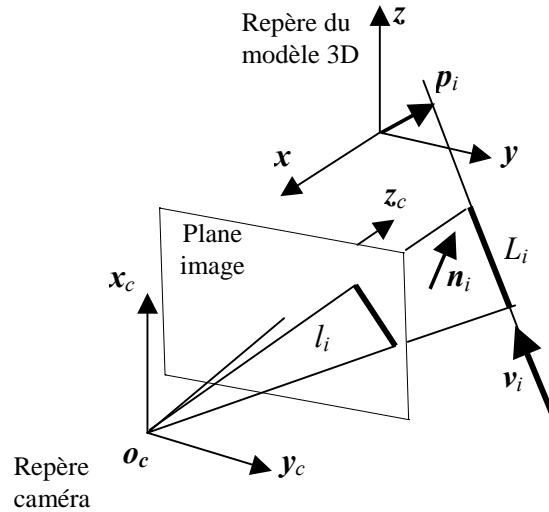
Considérant les coordonnées  $(x_c, y_c, z_c)$  dans le repère caméra de  $\mathbf{M}$ , on peut écrire

$$\begin{cases} u = \alpha_u \cdot \frac{x_c}{z_c} + u_0 \\ v = \alpha_v \cdot \frac{y_c}{z_c} + v_0 \end{cases} \quad (3.2.22)$$

En remplaçant  $u$  et  $v$  dans l'équation de la droite on obtient une équation de la forme

$a'x_c + b'y_c + c'z_c = 0$  qui donne sous forme vectorielle  $\overline{\mathbf{n} \cdot \mathbf{O}_c \mathbf{M}} = 0$  (figure 3.7).

Ce raisonnement étant valable pour tout point de  $L$ , le vecteur  $\mathbf{n}[a', b', c']^T$  n'est autre que la normale au plan d'interprétation contenant à la fois les droites  $L$  et  $l$  et le centre optique origine du repère caméra  $\mathbf{O}_c$ .



**Figure 3.7 :** projection d'un segment 3D

Le vecteur  $\mathbf{n}$  peut être calculé uniquement à partir de données image du segment  $l$  et connaissant les paramètres intrinsèques de la caméra. Chaque correspondance fournit deux contraintes pouvant se traduire par les produits scalaires théoriquement nuls entre  $\mathbf{n}$  d'une part et  $\mathbf{V}$  et  $\mathbf{P}$  d'autre part :

$$\mathbf{n} \cdot \mathbf{v} = \mathbf{n} \cdot (\mathbf{R}\mathbf{V}) = 0 \quad (3.2.23)$$

$$\mathbf{n} \cdot \mathbf{p} = \mathbf{n} \cdot (\mathbf{R}\mathbf{P} + \mathbf{t}) = 0 \quad (3.2.24)$$

Ces deux contraintes peuvent alors être utilisées pour exprimer une fonction d'erreur et résoudre le système par la méthode des moindres carrés non linéaires. Dans ce cas la fonction d'erreur s'écrit :

$$f(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^n (\mathbf{n}_i \cdot (\mathbf{R}\mathbf{V}_i))^2 + (\mathbf{n}_i \cdot (\mathbf{R}\mathbf{P}_i + \mathbf{t}))^2 \quad (3.2.25)$$

Chaque correspondance d'une droite de l'image avec un segment du modèle 3D fournit deux équations. Il faut donc un minimum de trois correspondances pour recouvrir les six paramètres permettant de localiser la caméra.

Il est à noter qu'avec cette approche, il est possible de découpler complètement la translation de la rotation. En effet, la première contrainte est une fonction de la rotation seulement. On peut donc calculer l'orientation de la caméra dans un premier temps en minimisant la fonction d'erreur suivante :

$$f(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^n (\mathbf{n}_i \cdot (\mathbf{R}\mathbf{V}_i))^2 \quad (3.2.26)$$

Une fois la rotation connue, le problème de la translation devient linéaire. En remplaçant  $\mathbf{R}$  dans la deuxième équation du système (3.2.24) par le résultat de la minimisation de (3.2.26), on obtient en effet un système d'équations linéaire par rapport à  $\mathbf{t}$ .

L'intérêt du découplage, est que l'on remplace la résolution d'un système à six inconnues par la résolution de deux systèmes à trois inconnues chacun. En effet, le coût en calcul lors de l'optimisation numérique de fonctions avec un nombre d'inconnues élevé peut être considérable. D'un autre côté, le fait de considérer  $\mathbf{R}$  comme constant lors de la minimisation de la fonction d'erreur concernant la translation introduit une incertitude supplémentaire. Le résultat peut être moins précis que l'optimisation de l'ensemble des paramètres en même temps.

Toute la difficulté réside dans le formalisme utilisé pour exprimer  $\mathbf{R}$ . Horaud propose une solution utilisant les quaternions pour représenter  $\mathbf{R}$  ([Horaud93]). La fonction d'erreur s'exprime comme suit :

$$f(\mathbf{r}) = \sum_{i=1}^n \left( (\mathbf{r}^t A_i \mathbf{r})^2 + (\mathbf{r}^t B_i \mathbf{r} + \mathbf{n}_i^t \mathbf{t})^2 \right) + \lambda (\mathbf{r}^t \mathbf{r} - 1)^2 \quad (3.2.27)$$

où  $\mathbf{r}$  est le quaternion unitaire de  $\mathbf{R}$ ,  $\lambda$  une constante positive,  $A_i = \mathbf{Q}(\mathbf{n}_i)^t \mathbf{W}(\mathbf{V}_i)$  et  $B_i = \mathbf{Q}(\mathbf{n}_i)^t \mathbf{W}(\mathbf{P}_i)$ . Les matrices  $\mathbf{Q}$  et  $\mathbf{W}$  sont définies comme suit :

$$\mathbf{Q}(\mathbf{x}) = \begin{bmatrix} 0 & -x_1 & -x_2 & -x_3 \\ n_1 & 0 & -x_3 & -x_2 \\ n_2 & x_3 & 0 & -x_1 \\ n_3 & -x_2 & -x_1 & 0 \end{bmatrix}, \quad \mathbf{W}(\mathbf{x}) = \begin{bmatrix} 0 & -x_1 & -x_2 & -x_3 \\ x_1 & 0 & x_3 & -x_2 \\ x_2 & -x_3 & 0 & x_1 \\ x_3 & x_2 & -x_1 & 0 \end{bmatrix}$$

Dans le cas de la solution où la rotation et la translation sont découplées, la fonction d'erreur pour le calcul de la rotation est :

$$f(\mathbf{r}) = \sum_{i=1}^n (\mathbf{r}^t A_i \mathbf{r})^2 + \lambda (\mathbf{r}^t \mathbf{r} - 1)^2 \quad (3.2.28)$$

Pour minimiser cette fonction, l'auteur utilise une méthode dite de région de confiance ([Phong95]). Cette méthode possède des propriétés de convergence moins dépendantes de la qualité d'initialisation. Cette caractéristique est importante compte tenu de la forte non-linéarité de la fonction d'erreur.

## Formulation adaptée à la robotique mobile

De la même façon que pour l'approche de Lowe, nous allons exprimer les deux contraintes dans le cas à trois degrés de liberté en adoptant l'expression (3.2.12) pour la rotation et en exprimant la translation dans le repère caméra.

### Fonction d'erreur

En remplaçant  $\mathbf{R}$  par son expression les contraintes (19,20) deviennent

$$\begin{cases} A \cdot \cos(\phi) + B \cdot \sin(\phi) + C = 0 \\ D \cdot \cos(\phi) + E \cdot \sin(\phi) + F \cdot y_c + G \cdot z_c + H = 0 \end{cases} \quad (3.2.29)$$

En posant encore une fois comme vecteur-inconnue  $\mathbf{q} = [\cos(\phi) \ \sin(\phi) \ y_c \ z_c]^T$  la fonction d'erreur devient

$$f(\mathbf{r}) = \sum_{i=1}^n \left( (A_i \cdot q_1 + B_i \cdot q_2 + C_i)^2 + (D_i \cdot q_1 + E_i \cdot q_2 + F_i \cdot q_3 + G_i \cdot q_4 + H_i)^2 \right) + \lambda (q_1^2 + q_2^2 - 1)^2 \quad (3.2.30)$$

avec

$$\begin{aligned} A_i &= (n_{1i} - n_{3i}) \cdot V_{1i} \cdot \cos(\theta) + n_{2i} \cdot V_{2i} \\ B_i &= n_{2i} \cdot V_{1i} - (n_{1i} - n_{3i}) \cdot V_{2i} \cdot \sin(\theta) \\ C_i &= n_{1i} \cdot V_{3i} \cdot \sin(\theta) + n_{3i} \cdot V_{3i} \cdot \sin(\theta) \\ D_i &= (n_{1i} - n_{3i}) \cdot P_{1i} \cdot \cos(\theta) + n_{2i} \cdot P_{2i} \\ E_i &= n_{2i} \cdot P_{1i} - (n_{1i} - n_{3i}) \cdot P_{2i} \cdot \sin(\theta) \\ F_i &= n_{2i} \\ G_i &= -\cot(\theta) \cdot n_{1i} + n_{3i} \\ H_i &= n_{1i} \cdot \left( P_{3i} \cdot \sin(\theta) - \frac{z}{\sin(\theta)} \right) + n_{3i} \cdot P_{3i} \cdot \sin(\theta) \end{aligned}$$

## Minimisation de la fonction d'erreur

### Solution non linéaire

Comme pour la méthode de Lowe, le terme  $\lambda(q_1^2 + q_2^2 - 1)^2$  est ajouté pour assurer l'orthogonalité de la matrice de rotation. Avec au moins deux correspondances de droites le système d'équations obtenu est surdéterminé et peut être résolu par la méthode de Gauss-Newton ou de Levenberg-Marquardt. Le calcul des dérivées partielles des termes de l'équation (3.2.30) par rapport à  $q_1$ ,  $q_2$ ,  $q_3$  et  $q_4$  est trivial.

En effet les dérivées des termes de la forme  $A_i \cdot q_1 + B_i \cdot q_2 + C_i$  sont

$$\begin{aligned} \frac{\partial(A_i \cdot q_1 + B_i \cdot q_2 + C_i)}{\partial q_1} &= A_i \\ \frac{\partial(A_i \cdot q_1 + B_i \cdot q_2 + C_i)}{\partial q_2} &= B_i \\ \frac{\partial(A_i \cdot q_1 + B_i \cdot q_2 + C_i)}{\partial q_3} &= 0 \\ \frac{\partial(A_i \cdot q_1 + B_i \cdot q_2 + C_i)}{\partial q_4} &= 0 \end{aligned}$$

Les dérivées des termes de la forme  $D_i \cdot q_1 + E_i \cdot q_2 + F_i \cdot q_3 + G_i \cdot q_4 + H_i$  sont

$$\begin{aligned} \frac{\partial(D_i \cdot q_1 + E_i \cdot q_2 + F_i \cdot q_3 + G_i \cdot q_4 + H_i)}{\partial q_1} &= D_i \\ \frac{\partial(D_i \cdot q_1 + E_i \cdot q_2 + F_i \cdot q_3 + G_i \cdot q_4 + H_i)}{\partial q_2} &= E_i \\ \frac{\partial(D_i \cdot q_1 + E_i \cdot q_2 + F_i \cdot q_3 + G_i \cdot q_4 + H_i)}{\partial q_3} &= F_i \\ \frac{\partial(D_i \cdot q_1 + E_i \cdot q_2 + F_i \cdot q_3 + G_i \cdot q_4 + H_i)}{\partial q_4} &= G_i \end{aligned}$$



Il est également possible d'exploiter la possibilité de découplage de la rotation et de la translation. Ceci revient à recouvrir dans un premier temps l'angle d'orientation en résolvant le système d'équations formé par la première contrainte de chaque appariement. Autrement dit en minimisant la fonction d'erreur suivante :

$$f(\mathbf{r}) = \sum_{i=1}^n (A_i \cdot q_1 + B_i \cdot q_2 + C_i)^2 + \lambda (q_1^2 + q_2^2 - 1)^2 \quad (3.2.31)$$

Le résultat est ensuite injecté dans l'équation de la seconde contrainte. Le système ainsi obtenu est linéaire et peut être résolu de la même façon que pour le système de la section suivante.

### Solution linéaire

Les avantages d'une solution linéaire sont l'affranchissement de l'initialisation et la réduction du coût calculatoire. Pour ce faire, une autre façon de procéder est de considérer les termes  $q_1$  et  $q_2$  comme indépendants. Le système d'équations précédent devient alors linéaire par rapport à  $\mathbf{q}$  et s'écrit

$$\mathbf{A} \cdot \mathbf{q} + \mathbf{B} = \mathbf{e} \quad (3.2.32)$$

avec

$$\mathbf{A} = \begin{bmatrix} A_1 & B_1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A_N & B_N & 0 & 0 \\ D_1 & E_1 & F_1 & G_1 \\ \vdots & \vdots & \vdots & \vdots \\ D_N & E_N & F_N & G_N \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} C_1 \\ \vdots \\ C_N \\ H_1 \\ \vdots \\ H_N \end{bmatrix} \text{ et } \mathbf{e} \text{ un vecteur d'erreur.}$$

La solution de ce système dans le sens des moindres carrés est

$$\mathbf{q} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (3.2.33)$$

Dans ce cas, les termes  $q_1$  et  $q_2$  sont calculés à un facteur d'échelle près qui est  $\frac{1}{(q_1^2 + q_2^2)}$ .

### Solution à 4 degrés de liberté

Le fait de réduire le nombre de degrés de liberté permet d'une part de diminuer les effets de la non-linéarité des équations à résoudre en terme de sensibilité par rapport aux conditions initiales et au bruit et en terme de temps de convergence. Le nombre minimal d'appariements nécessaire au calcul de la position et de l'orientation de la caméra est également réduit d'une unité. Cependant, dans une application réelle ceci introduit une nouvelle source de bruit dans les équations dérivées de chaque appariement. En effet, les trois rotations élémentaires définissant l'orientation de la caméra sont liées par la contrainte d'orthogonalité de la matrice de rotation. Il s'en suit que l'erreur commise sur les valeurs des deux angles supposés connus  $\theta$  et  $\psi$  influe sur le résultat du calcul de  $\phi$ . Ces erreurs sont dues aux incertitudes lors du calibrage du système base mobile et caméra mais aussi aux vibrations durant le mouvement et aux imperfections du plan horizontal sur lequel se déplace le robot. Des tests ont montré que cela est particulièrement le cas en ce qui concerne l'angle de site du robot qui détermine la valeur de l'angle  $\theta$ . Les performances de la méthode à trois degrés de liberté en terme de précision de localisation peuvent donc s'en ressentir.

Une solution consiste alors à faire un compromis entre ces deux méthodes extrêmes (à six degrés de liberté et à trois degrés de liberté) en ajoutant l'angle  $\theta$  en tant qu'inconnue dans le système à trois degrés de liberté. On obtient ainsi un système à quatre degrés de liberté. La fonction d'erreur à minimiser découlant de l'équation de contrainte (3.2.23) devient

$$f(\phi, \theta) = \sum_{i=1}^n (A_i \cos \theta \cos \phi + B_i \sin \theta \cos \phi + C_i \sin \phi + D_i \sin \theta + E_i \cos \theta + F \sin \phi \cos \theta + G_i \sin \theta \sin \phi + H_i \cos \phi)^2 \quad (3.2.34)$$

avec

$$\begin{aligned} A_i &= n_{1i} V_{1i} & E_i &= n_{2i} V_{2i} \\ B_i &= -n_{1i} V_{2i} & F_i &= -n_{3i} V_{1i} \\ C_i &= n_{1i} V_{3i} & G_i &= n_{3i} V_{2i} \\ D_i &= n_{2i} V_{1i} & H_i &= n_{3i} V_{3i} \end{aligned}$$

Le calcul de la translation reste inchangé par rapport à la méthode à trois degrés de liberté.

### Cas singuliers

Lorsqu'on utilise des méthodes numériques il est conseillé de surdéterminer le système pour un résultat plus précis. En d'autres termes il faut plus d'équations que d'inconnues. Dans le cas général, il faut un minimum de deux appariements pour résoudre les différents systèmes avec les précédentes méthodes. Lorsqu'un segment du modèle 3D est vertical, son vecteur directeur  $V$  est de la forme  $[0 \ 0 \ V_3]^T$ . Les constantes  $A$  et  $B$  du système (3.2.29) s'annulent. La première équation de ce système n'est alors plus utilisable. Un tel appariement fournit donc une seule équation et non deux comme dans le cas général. En conclusion, si tous les appariements utilisés sont formés de segments 3D verticaux, alors le nombre minimum d'appariements requis est de trois pour la solution non linéaire et quatre pour la solution linéaire.

## Evaluation des méthodes développées

Dans les chapitres précédents nous avons présenté plusieurs alternatives pour résoudre le problème du calcul de la position et de l'orientation d'une caméra avec une seule image et grâce à des correspondances de droite. Quelques avantages et faiblesses de chaque méthode ont été abordés. Une évaluation expérimentale est cependant nécessaire pour déterminer quelle méthode est la plus adéquate pour une application spécifique et dans des conditions spécifiques. Dans cette section nous présentons l'évaluation comparative de ces méthodes sur des données synthétiques et sur quelques cas avec des images réelles.

Le but des tests est d'évaluer d'une part l'applicabilité des méthodes présentées et ainsi formulées, et d'autre part, la comparaison de leurs performances, et ce dans un contexte particulier qui est celui de la localisation d'un robot mobile. Les tests ont donc volontairement été axés sur des configurations qui peuvent le plus souvent être rencontrées dans ce contexte en termes de domaine de variation des positions mais aussi des topologies des segments appariés utilisés. Les quatre méthodes évaluées avec leurs notations respectives sont

- La méthode basée sur l'approche Phong-Horaud à 3 degrés de liberté : PH3D,
- La méthode basée sur linéaire l'approche Phong-Horaud à 3 degrés de liberté : PH3D-lin,
- La méthode basée sur l'approche Phong-Horaud à 4 degrés de liberté : PH4D,
- La méthode basée sur l'approche de Lowe à 3 degrés de liberté : L3D.

## Données synthétiques

L'intérêt des batteries de tests utilisant des images synthétiques est qu'elles permettent l'évaluation des algorithmes d'un point de vue statistique avec un grand nombre de données couvrant un maximum de cas de figure possibles. Les positions de référence étant connus avec précision, les résultats peuvent être mieux analysés.

## Protocole

Un modèle synthétique composé de segments de droites représentant un environnement intérieur est construit. Grâce à un programme de simulation de caméra, des images de ce modèle sont générées à partir d'une position et d'une orientation donnée. La correspondance entre les segments de l'image et ceux du modèle étant connue, la position et l'orientation est calculées grâce à chacune des quatre méthodes étudiées.

Les caractéristiques du simulateur de caméra sont :

- résolution de 640x480,
- paramètres internes :  $u_0=240$ ,  $v_0=320$ ,  $\alpha_u=-1000$ ,  $\alpha_v=1000$ ,

Le modèle est composée de 50 segments. Les scènes sont construites grâce à un générateur de positions et d'orientations aléatoires. Les domaines de variations des paramètres de position et d'orientation sont :

- angle de site variant de  $-20^\circ$  à  $20^\circ$ , ce qui donne une variation de  $70^\circ$  à  $110^\circ$  pour  $\theta$ ,
- angle d'azimut variant de  $0^\circ$  à  $360^\circ$  ce qui donne une variation identique à l'angle  $\phi$ ,
- angle de lacet  $=0^\circ$ ,
- $x$  et  $y$  variant tous deux de 0m à 4m (couvrant ainsi l'espace navigable du modèle),

-  $z = 0.75\text{m}$ .

Toutes les données sont ensuite bruitées avant de générer les images synthétiques.

Un bruit additif a été ajouté aux angles  $\theta$  et  $\psi$  par le biais de deux variables aléatoires  $\varepsilon_\theta$  et  $\varepsilon_\psi$  de moyenne nulle et d'écart types  $\eta_\theta$  et  $\eta_\psi$  respectivement.

Pour modéliser les erreurs de calibrage, le modèle de caméra a été biaisé par ajout d'un facteur de bruit multiplicatif aux paramètres dépendant de la distance focale  $\alpha_u$ ,  $\alpha_v$  et d'un facteur de bruit additif aux coordonnées du centre de l'image  $u_0$ ,  $v_0$ . Un deuxième bruit additif a été ajouté pour modéliser les erreurs des traitement de détection et de localisation des primitives de l'image. Au final les coordonnées image des points composant les segments du modèle sont calculées suivant la formule suivante

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} (1 + \varepsilon_m)\alpha_u \frac{x}{z} + u_0 + \varepsilon_a \\ (1 + \varepsilon_m)\alpha_v \frac{x}{z} + v_0 + \varepsilon_a \end{bmatrix}$$

$\varepsilon_m$  et  $\varepsilon_a$  sont respectivement les paramètres de bruit multiplicatif et additif. Ils appartiennent tous deux à une distribution normale de moyenne nulle et d'écart type  $\eta_m$  et  $\eta_a$ .

Les paramètres  $\rho$  et  $d$  de chaque segment sont calculés comme suit

$$\begin{bmatrix} \rho \\ d \end{bmatrix} = \begin{bmatrix} \rho + \varepsilon_\rho \\ d + \varepsilon_d \end{bmatrix}$$

$\varepsilon_\rho$  et  $\varepsilon_d$  sont deux bruits Gaussiens additifs de moyenne nulle et d'écart type  $\eta_\rho$  et  $\eta_d$  respectivement.

Pour chaque scène, un vecteur de paramètres de position et de translation initial est généré de façon aléatoire. La qualité d'une initialisation est définie par la norme de la différence entre le vecteur recherché de translation (solution) et le vecteur initial de translation ainsi que par la différence entre l'orientation recherchée et l'orientation initiale. Un facteur de qualité noté  $Q_i$  est associé à chaque vecteur initial.

Dans les tests effectués, les valeurs retenues pour les paramètres de bruits sont regroupées dans la table 3.3. Les valeurs de  $\eta_m$ ,  $\eta_a$  sont les mêmes retenues dans [Araujo98] pour un tests similaire.

Paramètre(s) de bruit	Valeurs
$\eta_\psi$ et $\eta_\theta$	0 1 2 (°)
$\eta_m$	0.01 0.02 0.04
$\eta_a$	100 500 1000 (f)
$\eta_\rho$	0.5 1.0 2.0 (°)
$\eta_d$	1 2 3 (pixels)

**Table 3.3** : valeurs des paramètres de bruit utilisés dans le test 1

## Analyse des résultats

### Evaluation 1 : Précision de localisation et convergence dans le cas général

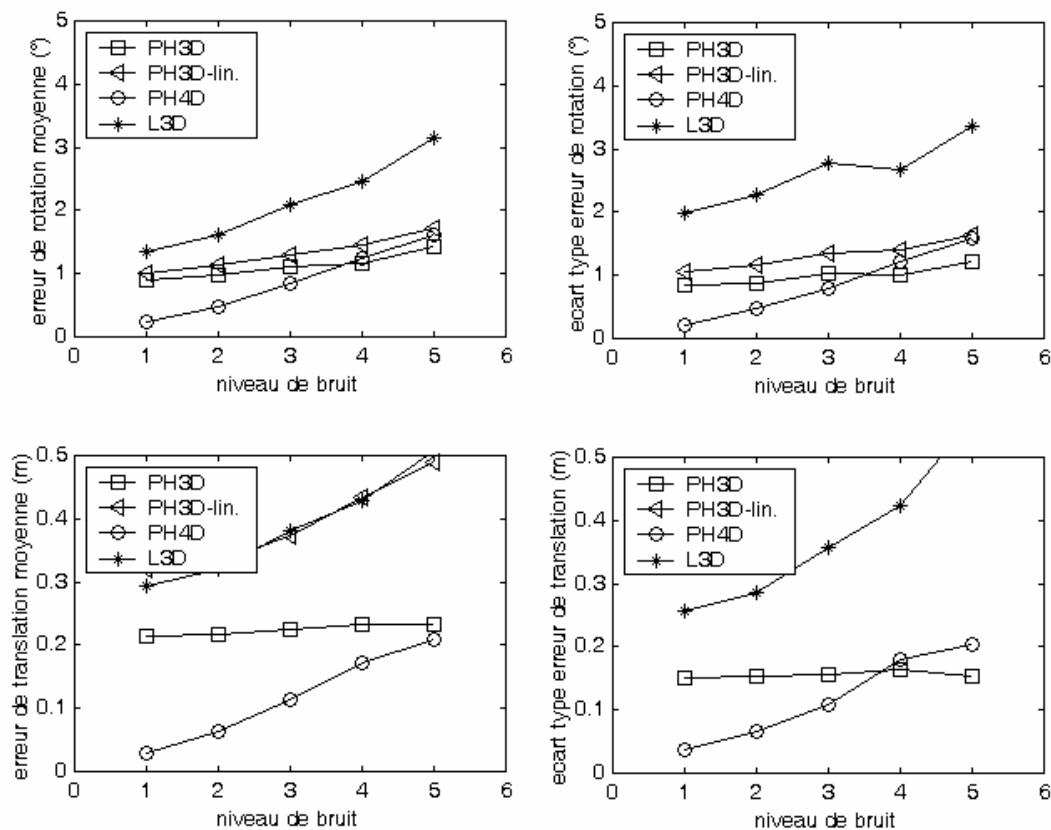
Pour chaque combinaison possible de  $\varepsilon_m$ ,  $\varepsilon_a$ ,  $\varepsilon_\rho$ ,  $\varepsilon_d$ ,  $\varepsilon_\theta$  et  $\varepsilon_\psi$  différentes scènes ont été générées. Les scènes où le nombre de segments visibles était inférieur à 3 ont été écartées. Au final 1200 images synthétiques avec un nombre de segments allant de 3 à 9 ont été retenues

pour chaque jeu de valeurs des paramètres de bruit. Les quatre méthodes de localisation ont été ensuite appliquées sur les mêmes scènes et avec les mêmes conditions initiales. Ce premier test concerne la précision de la localisation et sa sensibilité au bruit. Il a donc été réalisé avec de bonnes conditions initiales. Les paramètres d'orientation et de translation initiaux ont été générés de façon aléatoire pour chaque scène. L'orientation initiale était différente de  $5^\circ$  de l'orientation recherchée et la translation initiale différente de 50 cm de la solution. Les résultats obtenus sont représentés en figure 3.8. Les figures représentent en ordonnées les valeurs moyennes de l'erreur de localisation. L'erreur de rotation est la valeur absolue de la différence entre l'angle de référence  $\phi$  et l'angle calculé. L'erreur de translation est la norme de la différence entre le vecteur de translation de référence et le vecteur de translation calculé. Le niveau de bruit en abscisse représente les valeurs des paramètres de bruit utilisées lors de ce test. Ces valeurs sont données en table 3.4.

<i>valeur</i>	$\eta_\psi$ et $\eta_\theta$	$\eta_m$	$\eta_a$	$\eta_\rho$	$\eta_d$
1	2	0.01	0.01	0.5	1.0
2	2	0.02	0.02	1.0	1.0
3	2	0.02	0.02	1.0	1.0
4	2	0.04	0.04	2.0	3.0
5	2	0.04	0.04	2.0	3.0

**Table 3.4 :** *valeurs des paramètres concernant les résultats du test1*

La première remarque à faire est que la plupart des méthodes présentent peu de cas de singularité, c'est-à-dire des configurations pour lesquelles elles divergent. La seconde remarque est la rapidité de convergence. Sur les résultats des tests effectués, les trois méthodes non linéaires présentaient des nombres moyens d'itération pour converger très proches. Ces nombres se situaient entre 3 et 5 itérations. En ce qui concerne la précision de la localisation les méthodes basées sur l'approche de Phong-Horaud sont sensiblement plus précises et donnent en moyenne de meilleurs résultats que la méthode basée sur l'approche de Lowe. Elles sont également beaucoup moins sensibles au bruit. Parmi elles, la méthode à 4 degrés de liberté se détache du lot par une très bonne précision lorsque le niveau de bruit reste raisonnable.



**Figure 3.8:** résultats de la localisation des quatre méthodes en terme de précision de localisation

### Evaluation 2 : Sensibilité par rapport aux conditions initiales

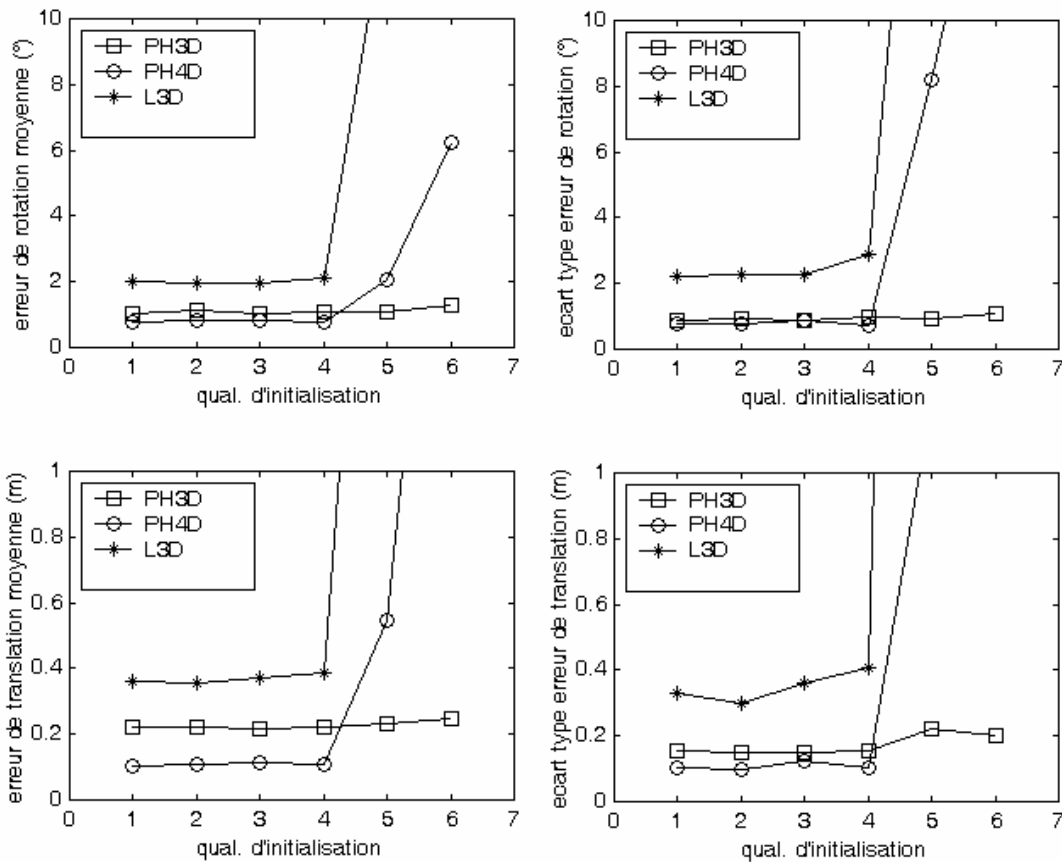
Le deuxième test concerne l'évolution des performances de chaque méthode en fonction de la qualité d'initialisation. Le test précédent a été répété avec un niveau de bruit constant mais en faisant varier la qualité d'initialisation. De nouveau 1200 scènes ont été générées et les trois méthodes non linéaires ont été appliquées sur toutes les scènes pour chaque niveau d'initialisation défini par les valeurs de la table 3.5. La figure 3.9 montre les résultats obtenus. Les abscisses représentent cette fois-ci la qualité de l'initialisation.

$Q_i$	Erreur d'orientation (°)	Erreur de translation (m)
1	5	0.20
2	10	0.40
3	15	0.80
4	20	1.00
5	30	1.20
6	45	1.40

**Table 3.5 :** initialisations utilisées pour le test 2

Il apparaît clairement que la méthode à quatre degrés de liberté, devient largement moins performante pour des initialisations moins bonnes. Ceci est dû à l'apparition de cas d'échec de la convergence ou encore de problèmes de minima locaux. La méthode Phong-Horaud non linéaire à 3 degrés de liberté reste à un niveau de performance stable et a un comportement

proche d'une méthode linéaire en terme de sensibilité aux conditions initiales. La méthode basée sur l'approche de Lowe quant à elle reste sensible à l'initialisation.



**Figure 3.9 :** performance de la localisation en fonction de la qualité d'initialisation

### Evaluation 3 : Sensibilité par rapport aux nombre d'appariements

Le troisième résultat concerne l'apport d'appariements supplémentaires à la précision de la localisation. La figure 3.10 montre l'évolution de l'erreur de localisation en fonction du nombre de segments appariés. Concernant l'orientation, on constate dans l'ensemble un gain en précision pour chaque appariement supplémentaire. Cette tendance est atténuée (voire inversée) pour le cas de la méthode de Phong-Horaud à 3 degrés de liberté. Cette dernière méthode est la plus performante lorsque le nombre d'appariements est réduit. La méthode de Lowe garde des performances stables à partir de 4 appariements. Les résultats de la translation montrent les mauvaises performances de l'approche de Lowe et de la méthode linéaire lorsque le nombre d'appariement est inférieur à 6.

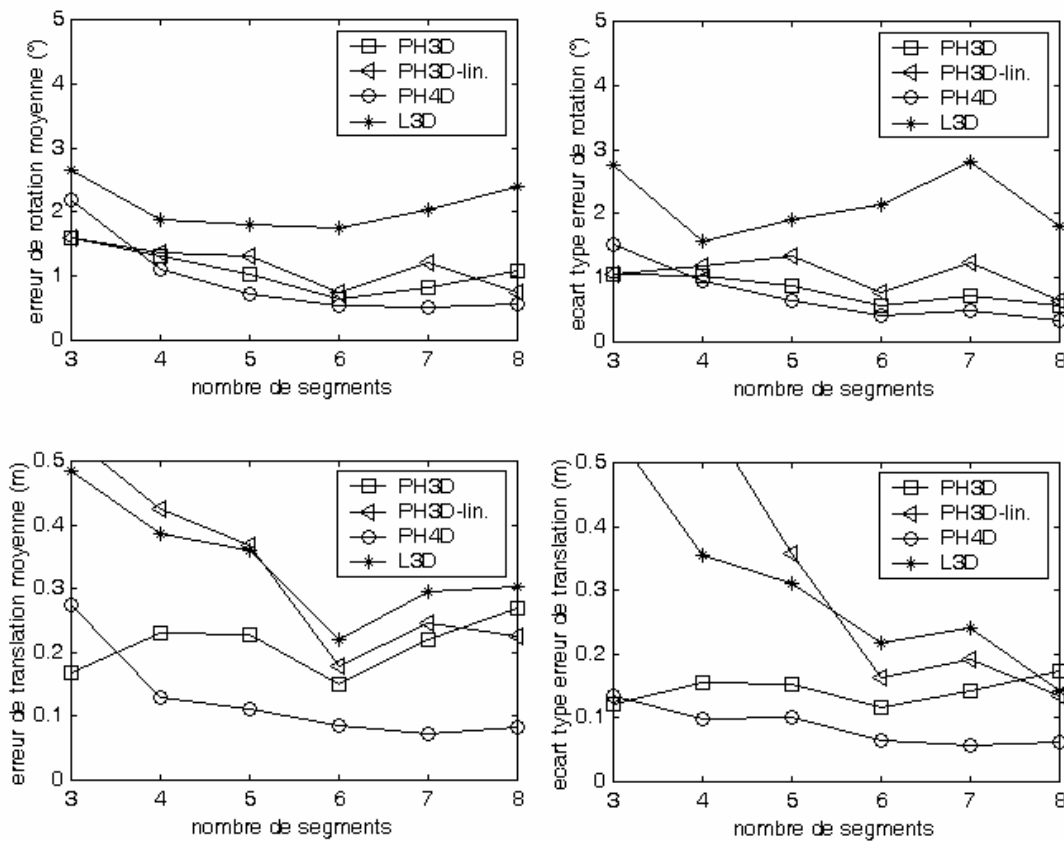


Figure 3.10 : évolution de l'erreur de localisation en fonction du nombre d'appariements utilisés

### Cas particuliers

Les résultats précédents concernent les performances dans le cas général. C'est-à-dire les cas où les scènes comportent suffisamment de segments dont la topologie (dans le modèle) ne présente pas des cas particuliers. En théorie les méthodes fonctionnent à partir de deux appariements non verticaux simultanément (sauf pour la méthode linéaire). En pratique, et pour que le système soit surdéterminé et donc mieux conditionné pour une optimisation dans le sens des moindres carrés, il faut au moins trois segments appariés. Les tests avec deux segments donnent environ 25 % de résultats erronés. Des tests spécifiques ont montré qu'avec des triplets de segments coplanaires et parallèles les résultats sont erronés dans à peu près 10% des cas étudiés.

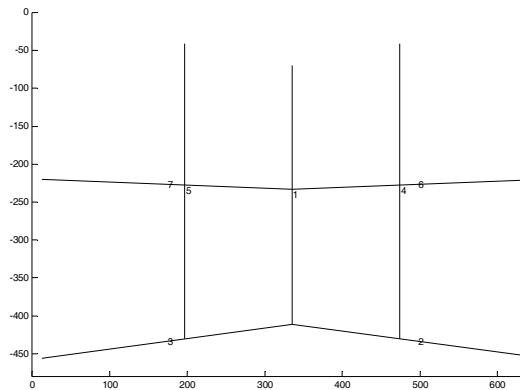
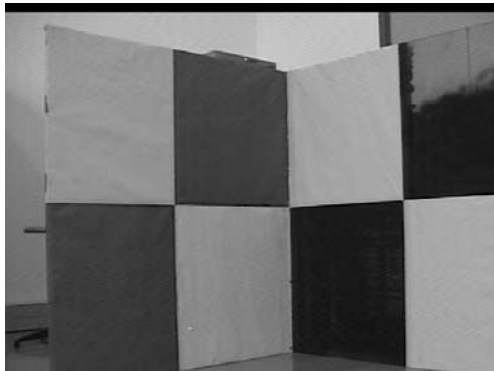
### Données réelles

#### Protocole

Les méthodes précédentes ont également été testées sur des images réelles. Un modèle simple représenté sur la figure 3.11 a été utilisé pour cela. Les tests ont porté sur plusieurs prises de vue avec différentes positions et orientations. Les positions de référence ont été mesurées sur



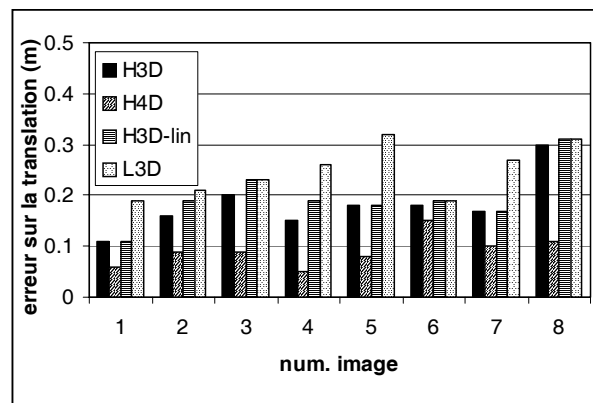
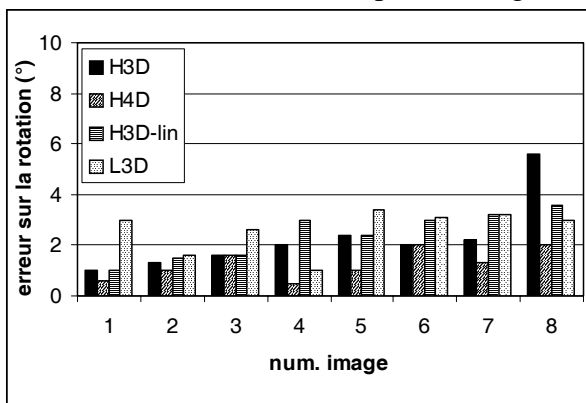
le terrain et calculées grâce à un calibrage avec la méthode de Faugeras-Toscani sur la mire de la figure 3.11. Des résultats représentatifs sont présentés sur les figures 3.12.a et 3.12.b.



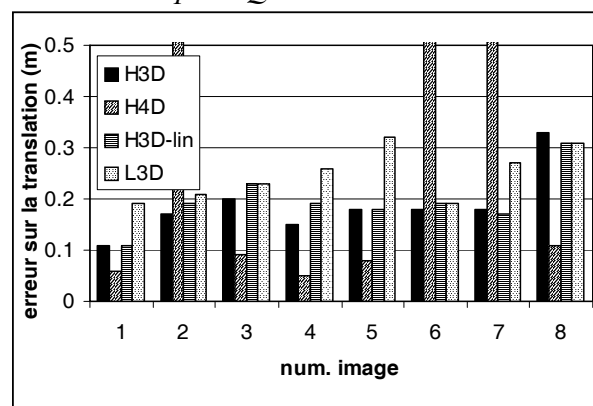
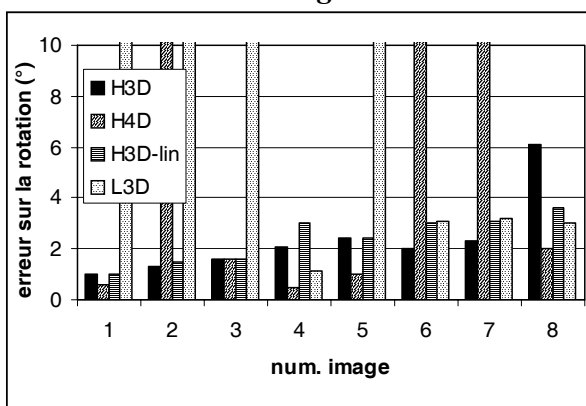
**Figure 3.11 :** modèle ayant servi à l'évaluation des méthodes de calcul de la position et de l'orientation sur des images réelles.

## Résultats

Les figures 3.12.a et 3.12.b montrent les erreurs de localisation obtenues avec deux qualités d'initialisation différentes ( $Q_i=1$  et  $Q_i=4$ ). Les orientations et positions de référence sont connues. Elles sont obtenues par calibrage.



**Figure 3.12.a :** erreur de localisation pour  $Q_i=1$



**Figure 3.12.b :** erreur de localisation pour  $Q_i=4$

Ces résultats confirment les tendances obtenues en simulation puisque il apparaît que la méthode basée sur l'approche de Phong-Horaud à quatre degrés de libertés semble la plus performante en terme de précision de localisation lorsque l'estimation initiale de la position et

de l'orientation est bonne ( $Q_i=1$ ). Cependant, des mauvais résultats apparaissent pour cette méthode lorsque l'estimation de la position et de l'orientation est plus large ( $Q_i=4$ ). Les méthodes à 3 DDL gardent un niveau d'erreur stable.

## Discussion

A la lumière des résultats de l'évaluation statistique avec les données synthétiques, un certain nombre de conclusions peuvent être tirées. Premièrement les méthodes basées sur l'approche Phong-Horaud sont généralement plus performantes que la méthode basée sur l'approche de Lowe. L'approche Linéaire est très sensible au bruit et donne de mauvais résultats lorsque le nombre d'appariements est réduit. La méthode à trois degrés de liberté donne un niveau d'erreur assez stable en fonction de la qualité d'initialisation et du nombre d'appariements. La méthode à quatre degrés de liberté est la plus performante en terme de précision et de robustesse par rapport au bruit. Cependant ses performances se détériorent lorsque les conditions initiales sont moins précises. Enfin, les quatre méthodes présentent également l'avantage de ne pas avoir de singularités pour des configurations telles que les jonctions de segments qui peuvent être rencontrées fréquemment dans un modèle d'environnement de type intérieur (coins). Les quelques résultats représentatifs obtenus avec des images réelles confirment ces tendances.

## Conclusions

Plusieurs méthodes de calcul de l'orientation et de la position à partir de correspondances de droites ont été présentées. Ces méthodes sont basées sur deux approches existantes qui sont l'approche de Lowe et l'approche de Phong-Horaud. L'apport principal de ce travail a été la dérivation d'expressions et d'équations spécifiques au problème de localisation de la caméra dans le cadre d'une application sur un robot mobile. Le but était de proposer des outils à même de répondre au mieux aux exigences d'une telle application en terme de rapidité, de sensibilité à la qualité de l'estimation initiale de la position ou encore de nombre de données nécessaires aux calculs.

Dans la méthode basée sur l'approche de Lowe, le vecteur de translation a été exprimé dans le repère caméra pour simplifier les expressions des équations de projections et ce en découplant certaines variables. Pour ne pas perdre l'avantage de la représentation dans un repère objet de cette translation qui est de permettre de réduire le nombre d'inconnues de translation à deux, une substitution des variables a permis néanmoins d'éliminer une des variables même dans cette représentation. Les expressions ainsi obtenues présentent donc les deux avantages de découplage et de réduction du nombre d'inconnues. Les variables de rotation sont directement celles composant l'orientation recherchée. Il n'est plus nécessaire de recomposer la matrice de rotation durant le processus de calcul comme dans l'approche originale.

Pour la méthode basée sur l'approche de Phong-Horaud, les simplifications ont abouti à deux solutions. Une solution linéaire qui permet de s'affranchir de la qualité d'initialisation et optimise le temps de convergence. Dans certains cas singuliers, cette solution n'est plus utilisable. Deux solutions non linéaires, une à quatre degrés de liberté et l'autre à trois degrés de liberté sont alors proposées. La première donne une localisation précise mais reste sensible à la qualité de l'estimation initiale de la solution. La seconde est moins précise mais a un comportement proche d'une solution linéaire en ce qui concerne l'initialisation. Les deux convergeant en quelques itérations seulement, une solution qui consiste à calculer l'estimation initiale par la méthode à trois degrés de liberté puis à affiner le résultat par celle à quatre degrés de liberté est une bonne stratégie en ligne. La faible sensibilité de la méthode à trois degrés de liberté aux variations du nombre d'appariements en fait une méthode appropriée au processus de mise en correspondance.

## Mise en correspondance 3D-2D

Dans la partie précédente, nous nous sommes intéressé au calcul de la position et de l'orientation de la caméra en supposant connus un certain nombre d'appariements entre les primitives de l'image et celles du modèle. Dans ce qui suit nous abordons le problème de la génération automatique de ce jeu d'appariements. C'est le problème de la mise en correspondance 2D-3D. C'est là sans doute l'étape la plus complexe dans les applications orientées vision. Après la présentation de quelques notions et termes nécessaires à la compréhension, nous présentons dans cette partie les principales approches existantes dans le domaine de la mise en correspondance puis la gestion de quelques problèmes inhérents à la présence de bruit dans les données. La méthode de mise en correspondance développée dans le cadre de cette thèse sera ensuite présentée en détails puis discutée.

# Etat de l'art

## Définitions

Le problème de la mise en correspondance, ou appariement, peut être posé comme suit : trouver, pour chaque élément d'un ensemble appelé modèle, l'élément correspondant dans un second ensemble appelé scène. En reconnaissance d'objets, par exemple, on dispose d'une base de données dans laquelle sont stockés des modèles d'objets et d'une vue du monde réel. La question est de dire si un objet de la base est présent dans cette vue et, si oui, dans quelle position et orientation il est par rapport à la caméra.

Soit  $\mathbf{M} = \{m_1, m_2, \dots, m_m\}$  l'ensemble de primitives géométriques composant le modèle 3D.

Soit  $\mathbf{S} = \{s_1, s_2, \dots, s_n\}$  l'ensemble de primitives géométriques composant la scène. Dans le cas idéal, le nombre de primitives du modèle  $m$  et le nombre de primitives de la scène  $n$  seraient égaux et il existerait une transformation  $F$  qui ferait coïncider les primitives du modèles avec celles de la scène. Le problème se réduirait alors à rechercher une permutation des index des primitives pour trouver celle qui produit la transformation  $F$  recherchée. En pratique, ce n'est pas aussi simple. Les primitives formant la scène n'appartiennent pas toutes à l'objet recherché. Certaines peuvent correspondre à des objets non modélisés ou à des défauts de détection. D'un autre côté, certaines primitives du modèle, sensées figurer sur la scène, n'y sont pas à cause d'occultations ou de mauvaises détections lors de l'acquisition de l'image ou lors de la segmentation. Il existe donc dans chaque ensemble des primitives qui n'ont pas à être appariées. Pour compliquer le tout, Les paramètres de mesure des primitives extraits de l'image et issus de la segmentation sont bruités. La superposition des primitives après application de la transformation  $F$  n'est donc jamais parfaite.

Désignons par  $\mathbf{II}$  la fonction d'appariement qui associe au  $i^{\text{ème}}$  élément de  $\mathbf{M}$  le  $j^{\text{ème}}$  élément de  $\mathbf{S}$  ou  $*$  (correspondant à une primitive non vue dans la scène).  $\mathbf{II}$  serait une application de  $\{1, 2, \dots, m\}$  vers  $\{1, 2, \dots, n, *\}$ . Il existerait donc  $(n+1)^m$  telles applications définissant l'espace des correspondances. On associe également à chaque appariement un facteur de qualité déterminé par la « distance » entre la transformée de la primitive du modèle et la primitive correspondante dans la scène. Cette distance dépend beaucoup de la fonction  $F$  qui appartient à l'espace des transformations. L'idée de base dans la mise en correspondance est de trouver la correspondance qui maximise le nombre d'appariements et, en même temps, la transformation qui minimise la distance entre les transformées des primitives appariées du modèle avec leur primitives correspondantes respectivement dans la scène.

## Méthodes existantes

Les méthodes de mise en correspondance peuvent généralement être regroupée en deux familles : les méthodes cherchant dans l'espace des correspondances et les méthodes cherchant dans l'espace des transformations. Nous présentons dans ce qui suit les principales méthodes développées dans ces deux catégories.

## Recherche arborescente

C'est une approche qui a pour principe de rechercher dans l'espace des correspondances. Soit  $\mathbf{M} = \{m_1, m_2, \dots, m_m\}$  l'ensemble de primitives géométriques composant le modèle 3D. Soit  $\mathbf{S} = \{s_1, s_2, \dots, s_n\} \cup \{*\}$  l'ensemble de primitives composant la scène. L'espace des correspondances est organisé sous une forme arborescente. Au sommet de cet arbre, dit arbre d'interprétation, un nœud représente la racine ou point de départ. Les appariements possibles de la primitive  $m_1$  du modèle avec les primitives de la scène sont ordonnés sous forme d'un premier niveau de nœuds reliés à la racine par des branches. Il existe donc  $n+1$  nœuds dans ce niveau. Chaque nœud de ce premier niveau est relié par des branches à un ensemble de nœuds représentant les appariements possibles de la primitive  $m_2$ . Ce deuxième niveau comprend donc  $(n+1)(n+1)$  nœuds. Un autre niveau est ajouté jusqu'à ce que toutes les primitives du modèle soient appariées (figure 4.1). La recherche consiste à explorer l'arbre en partant du point de départ pour trouver un chemin complet entre la racine et les feuilles. Ce chemin est ensuite validé ou éliminé en vérifiant si la transformation  $F$  correspondante obtient un bon score de qualité.

Le principal problème est alors l'explosion du nombre de correspondances (c'est-à-dire de chemins à explorer). Le nombre total de possibilités croît exponentiellement en fonction du nombre de primitives. Il est égal à  $(n+1)^m$ . Il est évident que la vérification de chaque combinaison en calculant la transformation correspondante serait fastidieux et incompatible avec une application réelle. La solution proposée par Grimson dans [Grimson87] et [Grimson90] consiste à élaguer l'arbre d'interprétation grâce à des contraintes géométriques. Les contraintes géométriques les plus utilisées sont les contraintes unaire ou binaire. Une contrainte unaire est une règle qui doit être vérifiée par un appariement d'une primitive du modèle avec une primitive de la scène. Un exemple classique est la longueur pour le cas où les primitives utilisées sont des segments de droites ([Muray88]). En effet, un segment vu sur l'image ne peut être plus long que le segment estimé à partir de son correspondant dans le modèle. Une contrainte binaire doit être satisfaite par l'appariement d'un couple de primitives du modèle avec un couple de primitives de la scène. Par exemple, l'angle estimé entre deux segments de droites de la scène doit être égal à l'angle entre les deux segments correspondants dans le modèle. Si un nœud ne satisfait pas une des deux contraintes ainsi définies, toutes les branches qui en découlent sont immédiatement élaguées réduisant le nombre de chemins à explorer.

Les applications les plus connues de cette approche ont été faites dans des cas où la scène et le modèle sont définis dans un même espace (appariement 2D-2D ou 3D-3D) ou alors dans des cas où les primitives de la scène 2D sont ramenées à un espace 3D par stéréovision. Une autre méthode présentée par Mallem ([Mallem96]) utilise la recherche arborescente pour trouver un ensemble d'appariements entre les segments de l'image et les arêtes d'objets polyédriques en contact avec un plan connu. Le nombre de degrés de liberté est ainsi réduit à 3. Pour améliorer la vitesse d'élagage de l'arbre d'interprétation, les arêtes en contact avec le plan sont placées en haut de l'arbre. Les paramètres 3D des segments de l'image sont obtenus par projection sur le plan de contact. Un coefficient de confiance est associé à chaque appariement hypothétique dans cette partie supérieure de l'arbre. Ce coefficient est calculé en fonction de l'orientation du segment dans l'image et de la différence de longueur entre les arêtes et les segments 3D projetées. Il est utilisé comme contrainte unaire pour l'élimination des appariements accidentels. La phase de vérification est basée sur la méthode de Dhome ([Dhome89]).

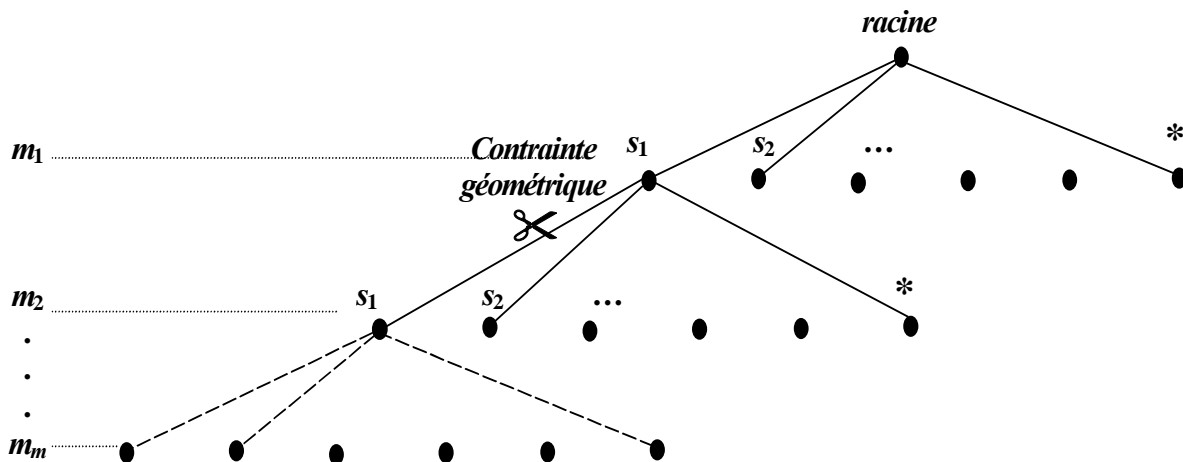


Figure 4.1 : construction de l'arbre d'interprétation

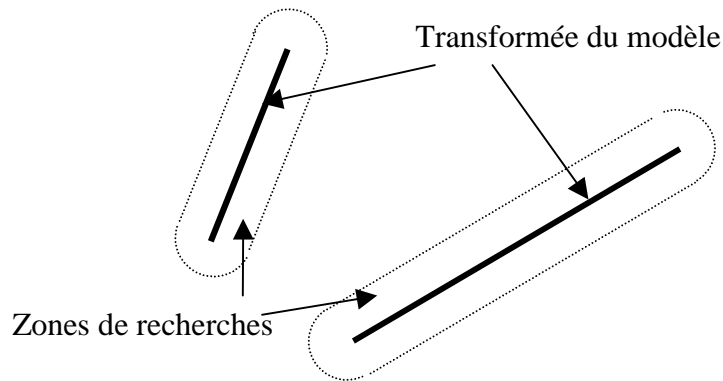
### Alignement ou prédiction vérification

Cette méthode s'exécute en deux étapes : premièrement les transformations possibles sont calculées en utilisant le nombre minimum de correspondances. Par exemple, pour une transformation dans un espace 3D, trois correspondances sont généralement nécessaires. Le nombre de possibilités est égal au produit du nombre d'arrangements des triplets de l'ensemble des primitives du modèle par le nombre de combinaisons de trois primitives de la scène soit

$$C_3^m C_3^n 3! \quad (4.1.1)$$

Les transformations correspondant aux différentes combinaisons de triplets de primitives du modèle avec les triplets de primitives de la scène sont calculées. En deuxième lieu, les transformations ainsi obtenues sont utilisées pour projeter toutes les primitives du modèle dans la scène. Les transformées obtenues sont alors utilisées pour rechercher de nouveaux appariements dans la scène. Une zone d'erreur est définie autour de chaque transformée (figure 4.2). Les paramètres des primitives correspondantes dans la scène sont donc « prédits » grâce aux transformations calculées sur la base d'un nombre minimum d'appariements. La correspondance est alors vérifiée en quantifiant le degré de coïncidence des données estimées et des données observées.

Pour rendre moins aléatoire le choix des appariements initiaux, certains auteurs ([Lowe87], [Mohan92]) utilisent la notion d'organisation perceptive des primitives de la scène. Fishler ([Fishler81]) choisit de vérifier immédiatement sur le reste des primitives chaque appariement initial hypothétique et d'arrêter la recherche dès qu'un consensus suffisant apparaît sur une transformation pour améliorer la rapidité de l'algorithme.



*Figure 4.2 : exemple de zone d'erreur pour des primitives segments*

## Extension des primitives focales

Cette méthode commence par la détection dans la scène de primitives focales. Autrement dit, de primitives caractérisées par leur unicité et la vraisemblance de sa détection. La détection est orientée par l'organisation perceptive des primitives de la scène. Une primitive focale détectée sert ensuite à prédire puis détecter dans son voisinage les primitives de la scène correspondant aux primitives du modèle voisines de la focale. L'ensemble des primitives est alors organisé en un graphe que l'on cherche à appairer avec les graphes de primitives du modèle. Cet appariement est fait par la recherche de cliques maximales entre les deux graphes. Un exemple de l'utilisation de cette approche est donné dans [Bolles82].

## Hachage géométrique

L'idée de base de cette méthode est de mémoriser off-line dans une table de hachage les informations invariantes par rapport aux transformations sur le modèle. Dans la phase de reconnaissance, des informations de la scène sont extraites avec la même représentation et comparées à la table de hachage. Pour extraire les invariants du modèle un sous ensemble de ses primitives est choisi comme base de référence ou repère. Les autres primitives sont alors exprimées dans ce repère. On considère plus tard un sous ensemble de primitives de la scène comme repère et l'on exprime les autres primitives dans ce repère. Dans l'hypothèse où les deux repères sont mis en correspondances, les coordonnées calculées dans chaque repère pour le reste des primitives devraient être les mêmes.

Prenons l'exemple d'un modèle défini par un ensemble de points coplanaires, soumis à une transformation de rotation et translation et voire même une mise à l'échelle. Deux points suffisent à définir un repère orthogonal. Dans la phase off-line, chaque paire de points est utilisée pour définir un repère et les coordonnées des autres points sont calculées puis discrétisées dans ces repères. Toutes ses informations sont stockées dans la table de hachage. Dans la phase on-line, une paire de points est arbitrairement choisie dans la scène et les coordonnées des autres points sont calculées et discrétisées dans le repère de la paire de points choisie. Pour chaque coordonnée présente dans la table de hachage, voter pour le repère correspondant. Le repère qui obtient le meilleur score est sélectionné comme étant celui qui correspond à la bonne mise en correspondance.

Quelques travaux de référence sur cette méthode sont présentés dans [Lamdan88] et [Wolfson90].



## Transformée de Hough généralisée

Cette méthode effectue la recherche dans l'espace des transformations. A l'origine, la transformée de Hough a été définie pour détecter des primitives géométriques comme les droites ou les ellipses. Ces primitives sont définies par un jeu de paramètres donnés. Le processus est exécuté par accumulation des évidences dans cet espace paramétrique discrétisé. Le principe a été adapté à la reconnaissance de formes [Fischer92] en considérant comme espace paramétrique l'espace des transformations. L'espace des transformations est d'abord discrétisé sous forme d'une table à cellules (accumulateur). Ensuite pour chaque  $k$ -tuple de primitives du modèle apparié avec un  $k$ -tuple de primitives de la scène ( $k$  étant le nombre minimum de correspondances nécessaires pour calculer une transformation unique) une transformation est calculée et un vote est ajouté à la cellule correspondante. Après le test de toutes les combinaisons de  $k$ -tuples possibles, la cellule ayant recueilli le plus de votes est sélectionnée comme étant celle qui correspond à la bonne transformation. Les appariements qui ont participé au vote pour cette cellule compose la bonne mise en correspondance.

L'une des difficultés de cette approche est la discrétisation de l'espace des transformations. Une alternative est présentée dans [Breuel92] dans laquelle les transformations calculées sont mises dans une liste. Un algorithme de clustering est ensuite appliqué.

## Algorithme itératif du plus proche voisin

A partir d'une transformation initiale  $F_0$ , le principe est d'alterner l'optimisation des correspondances et des transformations. Pour la transformation  $F_k$ , chaque primitive  $s_i$  de la scène est appariée à la primitive  $m_j$  du modèle telle que  $F_i(m_j)$  soit le plus proche voisin de  $s_i$  parmi l'ensemble des transformées des primitives du modèle. Le processus s'arrête lorsque, pour plusieurs itérations successives, on obtient la même transformation et la même correspondance. Cette approche a été adoptée pour la mise en correspondance de deux ensembles de points dans [Besl92] et [Zhang94].

La principale faiblesse de cette méthode est qu'au départ, la recherche du plus proche voisin pour chaque primitive peut aboutir à des appariements intempestifs, à moins que la transformation initiale soit suffisamment proche de la solution. Un autre inconvénient est que l'approche recherche des appariements sur l'ensemble des primitives du modèle avec l'ensemble des primitives de la scène. Le calcul de la transformation et la recherche du plus proche voisin à chaque itération doit être extrêmement efficace. C'est pour ces raisons que cette approche est mieux adaptée pour être utilisée plutôt dans une phase de vérification après application de l'un des algorithmes ci-dessus.

## Méthode de recherche par contrainte de la position

Cette classe de méthodes de mise en correspondance a été développée par Breuel ([Breuel92]). L'idée de base est que chaque appariement d'une primitive de la scène avec une primitive du modèle impose à la caméra une zone de positions dans l'espace global de travail. L'ensemble des contraintes décompose donc l'espace des positions possibles en classes d'équivalence à l'intérieur desquelles un même ensemble d'appariements est considéré. Le processus de mise en correspondance consiste alors à chercher l'espace des classes d'équivalences pour trouver celle qui maximise le nombre de contraintes satisfaites.

Breuel utilise une méthode adaptative qui successivement divise l'espace des positions et écarte la partie correspondant aux cellules ne satisfaisant pas des contraintes. Les cellules non écartées sont divisées récursivement jusqu'à ce qu'elles soient assez réduites pour définir une position acceptable. Les résultats de Breuel montrent que la technique peut opérer en un temps de calcul linéairement proportionnel au nombre d'appariements possibles.

## **Prise en compte de l'erreur**

Les algorithmes de mise en correspondance présentés ci-dessus donnent de bons résultats dans le cas théorique où les mesures utilisées pour définir les paramètres des primitives de la scène et le calcul des transformations sont rigoureusement exacts. Comme nous l'avons vu plus haut, les déformations du système d'acquisition de la scène (telles que les distorsions de l'image dues à la caméra), les défauts de détection (traitement de l'image et extraction de contours) et les imperfections de la segmentation sous forme de primitives géométriques de plus haut niveau (approximation polygonale) sont autant de facteurs qui font que la méthode utilisée pour la mise en correspondance doit tenir compte de la présence de bruit dans les mesures représentant les primitives de la scène. Ceci est fait par l'introduction de seuils de tolérance lors de la sélection des correspondances. Pour la recherche arborescente, par exemple, il faut introduire ces seuils dans les équations définissant les contraintes géométriques. Pour la prédiction-vérification, l'erreur est prise en compte lors de la définition des zones de recherche d'appariements.

Dans ce qui suit nous nous intéressons aux différentes méthodes de prise en compte de l'erreur dans la mise en correspondance.

### **Modèle d'erreur bornée**

Ce modèle suppose l'existence d'un support compact de l'erreur et d'une marge supérieure du bruit sur la localisation des primitives de la scène. Ceci requiert que chaque primitive de la scène se trouve à une distance inférieure aux marges de tolérance prédéfinies de la transformée de la primitive correspondante du modèle. C'est le modèle utilisé par Grimson et Huttenlocher ([Grimson90], [Grimson94]) ou par Lamdan et Wolfson ([Lamdan91]) pour la propagation de l'erreur dans le calcul des invariants pour le hachage géométrique ou pour la définition des zones de recherche pour l'alignement.

Ce modèle permet d'éviter la non-détection d'appariements candidats mais la propagation des marges supérieures de l'erreur dans les expressions analytiques peut conduire à des zones d'erreur beaucoup trop importantes. Pennec ([Pennec98]) a illustré ce problème pour le cas des méthodes basées sur l'alignement et le hachage géométrique. Il trouve notamment que les marges d'erreur prédites sont dans le meilleur des cas deux fois plus grandes que celles observées statistiquement. De plus, la généralisation de cette approche est difficile. Le calcul pour chaque méthode est spécifique et dépend du type de primitives utilisé.

### **Modèle d'erreur probabiliste**

Ce modèle suppose que les primitives de la scène sont normalement et indépendamment distribuées autour de la position prédite. Les primitives sont donc vues comme étant la réalisation d'un vecteur aléatoire de paramètres. Le principe utilisé est alors la propagation du

moment d'ordre un de sa distribution. Une primitive  $m$  est donc représentée par le couple  $(\bar{m}, \Sigma_{mm})$  formé par son vecteur moyen et sa matrice de covariance.

Ce modèle est parfaitement adapté à l'introduction d'une estimation quantitative de la qualité d'un appariement. Cependant, une distribution Gaussienne est par définition à support infini. Il existe donc toujours une probabilité non nulle d'appariement de deux primitives quelconques. La solution à ce problème est l'utilisation d'un modèle d'erreur probabiliste bornée ([Faugeras93]). Ce modèle est en fait une combinaison des deux précédents. La zone de recherche de candidats à l'appariement autour d'une primitive  $(\bar{s}, \Sigma_{ss})$  est alors définie comme suit

$$Z_s(v) = \{z \in M / d_{s, \Sigma_{ss}}(z) < v\} \quad (4.1.2)$$

avec

$$d_{s, \Sigma_{ss}}(z) = \sqrt{(z - \bar{s})^T \Sigma_{ss}^{-1} (z - \bar{s})} \quad (4.1.3)$$

$d_{s, \Sigma_{ss}}(z)$  est la distance de Mahalanobis.  $v$  est un paramètre sans dimension qui est généralement plus facile à déterminer qu'un paramètre métrique dans le cas du modèle d'erreur bornée. La zone ainsi définie est un hyperellipsoïde centré sur la valeur moyenne de la primitive dans l'espace paramétrique considéré.

## Eviter les fausses détections positives

En prenant en compte l'erreur dans les algorithmes de mise en correspondance, on garantit de ne pas éliminer des appariements possibles. Cependant il existe aussi le risque que plusieurs primitives de la scène tombent accidentellement dans une zone de recherche. Ce qui peut amener à valider une correspondance « fantôme » et détecter un objet alors qu'il n'est pas présent dans la scène. La façon la plus utilisée pour résoudre ce problème est la mise en place lors de l'étape de vérification d'un seuil sur la fraction du nombre de primitives correctement appariées sur le nombre total de primitives en jeu dans une correspondance.

Un travail représentatif dans ce domaine est celui de Grimson ([Grimson91]). Dans son approche, l'espace des transformations est découpé en un certain nombre de cellules. Il calcule ensuite le volume couvert par l'appariement d'une primitive du modèle avec une primitive de la scène dans cette espace. Ce volume couvre un certain nombre de cellules. L'auteur considère ensuite chaque appariement comme un événement aléatoire à distribution uniforme. La répartition de l'ensemble des événements produits par la mise en correspondance de  $m$  primitives du modèle avec  $s$  primitives de la scène est alors modélisée par un modèle d'occupation de Maxwell-Boltzmann. Ceci permet de calculer la probabilité que  $k$  événements se produisent dans une même cellule. Autrement-dit, la probabilité que  $k$  volumes de l'espace des transformations aient une intersection commune. Cette probabilité qui a une expression de la forme :

$$p_k \approx \frac{\lambda^k}{(1 + \lambda)^{k+1}} \quad (4.1.4)$$

est ensuite utilisée pour définir la fraction minimale du nombre de primitives correctement appariées pour valider une correspondance en imposant que la probabilité qu'au moins  $k$  événements se produisent dans une même cellule soit inférieure à un seuil  $\delta$ :

$$1 - \sum_{k=0}^{mf_0-1} p_k \leq \delta \quad (4.1.5)$$

$f_0$  étant la fraction recherchée.

Le seuil obtenu est dynamique et dépend de la complexité de la scène et du modèle. La définition d'un seuil fixe serait compliquée et la valeur choisie ne serait pas adaptée à tous les points de vues.

D'autres travaux sur le problème des fausses détections sont présentés dans [Pennec98].

## Quelques travaux en robotique mobile

### Kosaka et Kak ([Kosaka92])

Dans le cadre du projet FINALE, la localisation utilise ici la représentation de la position et de l'orientation du robot par un modèle d'incertitude statistique. Pour faciliter la recherche d'appariements entre les primitives de l'image et celles du modèle, les incertitudes sur le vecteur de paramètres d'orientation et de position du robot  $\mathbf{p}=[\phi, x, y]^T$  sont propagées, grâce aux équations du mouvement et celles de la projection perspective, pour obtenir l'incertitude sur les coordonnées  $\mathbf{I}(u,v)$  des projections de chaque point  $\mathbf{M}$  du modèle sur l'image. Des zones de recherche d'appariements pour chaque point sont ainsi définies.

Soit  $\mathbf{p}$  le vecteur exprimant l'estimation de l'orientation et de la position du robot à un instant donné. Cette estimation est vue comme une variable aléatoire de valeur moyenne  $\bar{\mathbf{p}}$  et de matrice de covariance  $\Sigma_p$ . Soit  $\mathbf{h}$  la fonction qui définit l'évolution des paramètres de localisation suite à l'exécution d'une commande de mouvement. Le nouveau vecteur d'orientation et de position du robot après l'exécution de cette commande est donc  $\mathbf{p}_e=\mathbf{h}(\mathbf{p})$ . la nouvelle estimation de la localisation est représentée par le nouveau vecteur moyen  $\mathbf{p}_e$  et la nouvelle matrice de covariance  $\Sigma_{pe}$ . On a alors :

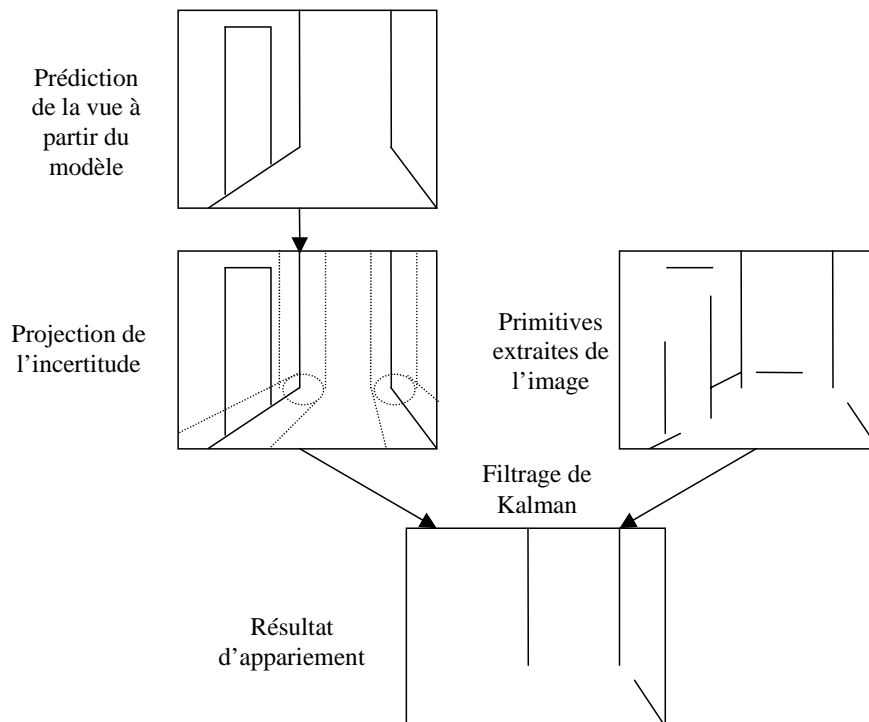
$$\bar{\mathbf{p}}' = \mathbf{h}(\bar{\mathbf{p}}) \quad (4.1.6) \quad \text{et} \quad \Sigma_{pe} = \mathfrak{S}_h \Sigma_p \mathfrak{S}_h^T \quad (4.1.7)$$

où  $\mathfrak{S}_h$  est la matrice jacobienne de  $\mathbf{h}$ .

Soit à présent  $\mathbf{H}$  la fonction qui définit la projection  $\mathbf{I}(u,v)$  sur l'image d'un point  $\mathbf{M}(x,y,z)$  de l'environnement en utilisant la position et l'orientation du robot  $\mathbf{p}'$ . On a de nouveau :

$$\bar{\mathbf{I}} = \mathbf{H}(\bar{\mathbf{p}}') \quad (4.1.8) \quad \text{et} \quad \Sigma_I = \mathfrak{S}_H \Sigma_{pe} \mathfrak{S}_H^T \quad (4.1.9)$$

Une zone de l'image est donc définie autour de la projection estimée par  $\bar{\mathbf{I}}$  et  $\Sigma_I$  grâce à la distance de Mahalanobis (figure 4.3). Les points de l'image avec lesquels  $\mathbf{M}$  peut être apparié sont donc à chercher dans cette zone. Ce raisonnement est étendu aux correspondances de segments de droite. Sur la base de ces appariements, un filtre de Kalman est ensuite utilisé pour la mise à jour des paramètres de position et d'orientation du robot. Les équations du filtre de Kalman sont obtenues grâce aux contraintes que doivent satisfaire les paramètres d'un segment de droite dans l'environnement et les paramètres d'une droite de l'image dans l'espace de Hough des paramètres de cette droite. Les équation  $\mathbf{h}$  et  $\mathbf{H}$  sont non-linéaires. Pour calculer les Jacobiennes, elles sont approximées par des fonctions linéaires en supposant un faible déplacement entre les mesures.



*Figure 4.3 : principe de recherche d'appariements dans le système FINALE*

## Talluri ([Talluri96])

Bien que ce travail ait comme plate forme d'essai l'environnement d'une aire d'aéroport, la nature structurée de cet environnement fait que la technique peut être utilisée comme méthode de localisation dans un environnement d'intérieur. En effet, les primitives utilisées dans ce travail sont les arêtes horizontales des toits d'immeubles.

L'espace navigable du robot est découpé en parties polygonales (Edge Visibility Regions) auxquelles sont associées des listes de segments visibles lorsque le robot se trouve à l'intérieur de l'une d'elles. Un appariement d'un segment de l'image avec un segment de l'environnement doit satisfaire une contrainte sur la position et l'orientation du robot dans l'espace de Hough de ces paramètres. L'accumulation des évidences permet alors de sélectionner une ou quelques régions candidates permettant de réduire le nombre d'appariements candidats aux seules listes de visibilité des régions retenues. La mise en correspondance est alors achevée par une recherche arborescente.

L'équation de contrainte est obtenue à partir des équations (3.1.16) et (3.1.17) de la méthode présentée dans le Chapitre 2. La réduction du nombre de degrés de liberté aux seules variables  $x$ ,  $y$  et  $\phi$  permet en effet de contraindre le domaine de variation de ces variables.

## Discussion

Nous avons présenté les principales solutions proposées au problème de la mise en correspondance dans ces dernières années. Nous avons vu également quelques problèmes parfois généraux tels que la prise en compte de l'erreur et l'évitement de fausses détections positives, d'autres fois spécifiques à certaines méthodes tels que le clustering pour la transformée de Hough ou la qualité d'initialisation pour la méthode du plus proche voisin. Dans un souci de concision, nous avons essayé de présenter les méthodes de façon générale sans aborder certaines spécificités comme le type de primitives utilisées ou les formes de

transformations considérées (transformation affine, 2D ou 3D, similitude,...), ce qui a limité la possibilité de donner des formulations mathématiques rigoureuses des problèmes. A la lumière de cette présentation il apparaît que le choix d'une technique de mise en correspondance pour notre application nécessite la formulation de tous les aspects du problème de la mise en correspondance présentés ici en fonction des caractéristiques de notre contexte, à savoir :

- les primitives sont des segments de droites,
- l'espace des transformations est à six dimensions avec une partie de paramètres connus,
- une estimation initiale de la transformation recherchée est disponible.

Analysons la façon dont se présentent les primitives dans des prises de vue d'environnement intérieur. Les segments observés correspondent le plus souvent à des fragments de segments du modèle compte tenu des grandes dimensions de ces derniers. Ceci rend difficile la constitution de repères image fiables. De plus les effets de perspective sont sensibles et rendent difficile le regroupement perceptif des primitives par parallélisme par exemple. Tous ces aspects nous font penser que des approches comme le hachage géométrique ou l'extension de primitives focales sont difficilement applicables. Il en va de même pour des approches de prédiction vérification à cause du nombre réduit de segments modèle qui peuvent être observés à un instant donné. L'approche de recherche arborescente, quant à elle, est dépendante des performances de sélection des contraintes géométriques locales utilisées pour l'élagage de l'arbre d'interprétation. Une bonne capacité d'élagage permet de contrecarrer l'explosion exponentielle du nombre d'hypothèses à traiter. En l'absence de capteur stéréoscopique qui permettrait d'estimer les paramètres 3D d'un segment de l'image, il faudrait que ces contraintes puissent être appliquées sur des appariements de données 2D et 3D.

Dans une approche comme celle du système FINALE ([Kosaka92]), l'efficacité de la stratégie de recherche d'appariements est liée à la précision de la position prédite du robot.

L'augmentation du niveau d'erreur peut conduire à étendre les zones de recherche autour de chaque segment projeté du modèle sur pratiquement toute l'image en raison des effets de la perspective et des dimensions des segments observés. La mise en correspondance est alors compromise. Cette approche suppose donc un niveau d'échantillonnage élevé pour maintenir un niveau d'erreur faible sur la prédiction de la position. L'aspect temps réel des calculs de localisation est alors très déterminant. D'un point de vue plus pratique, il faut aussi prendre en compte la disponibilité d'un nombre suffisant de données image et modèle pour effectuer le calcul à chaque prise de vue. En effet, il arrive qu'une acquisition d'image donnée ne fournisse pas suffisamment de primitives en raison de défauts de détection ou de segmentation. Il se peut également que le robot soit dans une position et une orientation telle que la liste des primitives potentiellement visibles du modèle soit trop réduite. Un autre facteur à prendre en compte est le fait que les systèmes robotisés sont généralement multitâche. Dans notre cas, la caméra sert également à la commande et à l'exploration. Il peut donc arriver que les ressources du système soit réquisitionnées au moment de l'acquisition de l'image. Dans tous ces cas, le calcul de localisation doit pouvoir être différé à une acquisition ultérieure, ce qui a, bien entendu, pour effet d'augmenter l'incertitude sur la prédiction de la position et de l'orientation.

L'approche de Talluri ([Talluri96]) semble appropriée mais elle est restreinte dans ces travaux à des segments horizontaux et est définie dans un contexte de localisation en milieu externe bien spécifique.

# Méthode développée

## Contexte de la mise en correspondance

Pour situer la problématique de la mise en correspondance dans le contexte de la robotique mobile, rappelons les caractéristiques importantes de ce domaine. Ces caractéristiques ont orienté ou déterminé les choix de techniques et de paramètres composant la solution proposée pour résoudre ce problème.

- Le modèle est constitué d'un ensemble  $\mathbf{L}=\{L_1, L_2, \dots, L_m\}$  de segments de droites 3D définis dans le repère modèle. La scène est représentée par l'ensemble  $\mathbf{l}=\{l_1, l_2, \dots, l_n\}$  de segments de droites 2D extraits d'une image prise par la caméra à partir de la position courante du robot. Certains correspondent aux projections des primitives du modèle, d'autres soit à des objets non modélisés présents dans la scène, soit à de fausses détections dues au bruit et aux erreurs de traitement d'image.
- La transformation reliant les deux repères, modèle et caméra, est à six variables puisque la position et l'orientation de la caméra sont définies dans un espace 3D (3 coordonnées pour la translation et 3 composantes d'orientation). Cependant trois sont connues (deux orientations élémentaires et une composante de la translation). Le vecteur des inconnues est donc à trois composantes  $\phi, x$  et  $y$ .
- Les dimensions des segments du modèle sont du même ordre de grandeur, voire même plus grand, que leurs distances au centre focal de la caméra. Les effets de la projection perspective sont donc considérables. L'utilisation d'un modèle perspectif de caméra est donc essentiel.
- Dans la majorité des cas, une estimation de la position est fournie soit par l'odométrie soit par la mémorisation des états précédents du robot et la propagation de l'incertitude en se basant sur les commandes de déplacement générées par le système. La qualité de cette estimation est variable selon sa source et la fréquence d'échantillonnage du calcul de localisation absolue. Dans ce qui suit, nous supposons que le système dispose d'une estimation de l'orientation  $\phi_e$  et d'une marge supérieure d'erreur sur cette orientation  $\Delta\phi_e$  ainsi que d'une estimation de la position  $[x_e, y_e]$  et d'une marge supérieure d'erreur de cette position  $[\Delta x_e, \Delta y_e]$ .

## Algorithme de mise en correspondance

L'algorithme développé pour la recherche de la bonne correspondance entre l'ensemble des segments de l'image et l'ensemble des segments constituant le modèle de l'environnement de travail est basé sur une approche de recherche arborescente. Il se décompose en trois phases principales :

- une phase de réduction de l'espace de recherche par sélection d'un sous-ensemble de primitives du modèle ;
- une phase de génération d'hypothèses de correspondances ;
- une phase de vérification et de sélection de la meilleure hypothèse ;

La première phase permet de réduire l'ensemble des segments candidats à l'appariement dans le modèle global en se basant sur une estimation de la position et de l'orientation du robot. Comme nous le verrons, la contrainte sur la précision de cette estimation est très flexible.

Cette étape a pour but, d'une part de réduire les effets de l'explosion exponentielle du nombre de combinaisons d'appariements, d'autre part de limiter les cas d'ambiguïtés où plusieurs combinaisons peuvent paraître cohérentes. En effet, à partir de sa position courante le robot ne peut percevoir qu'une vue locale du modèle, c'est-à-dire un sous-ensemble de primitives du modèle global. La sélection de ce sous-ensemble se fait grâce aux régions d'invariance visuelle qui seront abordées ci-dessous.

La seconde phase est celle de la génération d'hypothèses cohérentes de correspondance par l'élagage de l'arbre d'interprétation. Premièrement, les ensembles de primitives de l'image et du modèle sont chacun subdivisés en deux sous ensembles en tenant compte de la configuration du système et de la disposition des segments de l'image. On obtient ainsi deux arbres d'interprétation de dimensions inférieures au lieu d'un seul grand arbre. Deux contraintes, une unaire et l'autre binaire sont ensuite définies conformément aux caractéristiques du contexte citées plus haut. Elles sont directement applicables sur des appariements 2D-3D. Elles sont utilisées pour ne retenir qu'un nombre limité d'hypothèses de correspondance.

Enfin, la phase de vérification et de sélection consiste à calculer la transformation correspondant à chaque hypothèse retenue à l'issue de l'étape précédente et à appliquer cette transformation sur les segments du modèle. Une fonction permettant de quantifier le degré de vraisemblance de chaque hypothèse de correspondance sera également définie. L'hypothèse la plus vraisemblable est alors sélectionnée selon cette fonction. Ces différentes étapes sont détaillées dans les sections suivantes.

## **Sélection d'un sous-ensemble de primitives du modèle : régions d'invariance visuelle**

Le modèle 3D global peut comporter un nombre important de segments. Essayer de chercher la bonne mise en correspondance entre l'ensemble des segments d'une image et le modèle global serait fastidieux en raison du nombre trop important d'hypothèses à traiter. Un temps de calcul très important serait, dans ce cas, consacré à l'étude des faux appariements. Par exemple, il est inutile de chercher à apparier un couple de segments  $L_i-L_j$  du modèle avec des segments de l'image si l'on sait par avance qu'il n'existe pas, sur le plan de déplacement du robot, de position et d'orientation à partir desquelles  $L_i$  et  $L_j$  peuvent être observés en même temps. De plus, le risque d'erreur lors de la mise en correspondance du modèle global est plus important à cause des ambiguïtés.

Pour une image prise à un instant donné, on extrait un ensemble de segments qui constitue la scène. On dispose pour ce même instant d'une estimation de la position et de l'orientation de la caméra représentée par un vecteur  $[x_e, y_e, \theta_e]$  et d'une marge d'erreur pour chacun de ces paramètres  $[\Delta x, \Delta y, \Delta \theta]$ . On définit ainsi une zone d'erreur permettant de sélectionner une plusieurs RIV (Région d'Invariance Visuelle).

Si une seule RIV est sélectionnée, alors les segments appartenant à sa liste de visibilité sont retenus. Parmi ces segments on écarte les segments dont l'intervalle d'angle de visibilité associée ne coupe pas l'intervalle  $[\theta_e - \Delta \theta, \theta_e + \Delta \theta]$ . L'arbre d'interprétation est ainsi construit avec les segments restants.

Si plusieurs RIV sont retenues, l'opération précédente est répétée pour chacune des RIV. Un arbre est construit pour chaque région. Les hypothèses retenues à l'issue de l'élagage de chaque arbre sont finalement regroupées.

## **Génération des hypothèses de correspondances**



Dans la suite de ce chapitre, nous désignerons par  $\mathbf{L}=\{L_1, L_2, \dots, L_m\}$  le sous-ensemble des  $m$  segments du modèle sélectionnés grâce aux régions d'invariance visuelle.

Pour permettre l'élagage rapide de l'arbre d'interprétation issue des appariements de  $\mathbf{L}$  et  $\mathbf{I}$  nous définissons les contraintes géométriques nécessaires. Plus l'ordre de la contrainte utilisée est faible plus l'opération d'élagage est efficace. Nous commençons donc par définir une contrainte unaire. Nous verrons qu'elle présente une singularité lorsque les segments du modèle sont verticaux. Une contrainte binaire est alors définie pour pallier cette singularité. En tenant compte de la configuration de la caméra (notamment du fait que  $\psi=0$  et que  $\theta$  est connu), nous pouvons imposer des limites pour la pente  $\rho$  de chaque segment de l'image correspondant à un segment vertical dans le modèle. Au lieu d'adopter l'approche classique qui donnerait un arbre avec un nombre de cas égal à  $(n+1)^m$ , l'idée retenue consiste à subdiviser l'arbre global en deux sous-arbres : un arbre pour sélectionner grâce à la contrainte unaire les appariements des segments non verticaux du modèle et un autre pour sélectionner grâce à la contrainte binaire les segments verticaux. Le gain en terme de dimensions d'espace de correspondance à explorer sera discuté plus loin.

### Contrainte unaire

Soit  $L_i$  un segment du modèle de vecteur directeur  $V_i$  et de vecteur position  $P_i$ . Soit  $l_j$  un segment de la scène dont  $N_j$  est le vecteur normal au plan d'interprétation (figure 3.3). En reprenant les équations de contrainte (3.1.16) et (3.1.17) nous pouvons écrire

$$\begin{cases} a_{ij} \cdot \cos(\phi) + b_{ij} \cdot \sin(\phi) + c_{ij} = 0 & (4.2.1) \end{cases}$$

$$\begin{cases} A_{ij} \cdot x + B_{ij} \cdot y + C_{ij} = 0 & (4.2.2) \end{cases}$$

avec

$$\begin{cases} a_{ij} = (n_x \cos(\theta) - n_z \sin(\theta))v_x + n_y v_y \\ b_{ij} = n_y v_x - (n_x \cos(\theta) - n_z \sin(\theta))v_y \\ c_{ij} = (n_x \sin(\theta) + n_z \cos(\theta))v_z \\ A_{ij} = -(n_x \cos(\theta) - n_z \sin(\theta))\cos(\phi) - n_y \sin(\phi) \\ B_{ij} = -n_y \cos(\phi) + (n_x \cos(\theta) - n_z \sin(\theta))\sin(\phi) \\ C_{ij} = [(n_x \cos(\theta) - n_z \sin(\theta))x_i + n_y y_i] \cos(\phi) + [n_y x_i - (n_x \cos(\theta) - n_z \sin(\theta))y_i] \sin(\phi) + \\ [n_x \sin(\theta) + n_z \cos(\theta)](z_i - z) \end{cases}$$

(4.2.3)

Ceci signifie qu'avec un seul appariement nous pouvons calculer l'orientation  $\phi_{ij}$  de la caméra grâce à l'équation (4.2.1). Une fois cette orientation calculée, elle peut servir à contraindre la position à appartenir à une droite  $D_{ij}$  puisque l'équation (4.2.2) devient une équation de droite d'inconnues  $x, y$ . Si l'appariement considéré est bon alors l'orientation doit être incluse dans l'intervalle d'orientation défini par la marge d'erreur  $\Delta\phi_e$  autour de la valeur estimée  $\phi_e$ . De plus la droite définie grâce à l'équation (4.2.2) doit couper la zone du plan horizontal de déplacement définie par les marges d'erreur  $\Delta x_e$  et  $\Delta y_e$  autour de la position estimée  $[x_e, y_e]$  (figure 4.4).

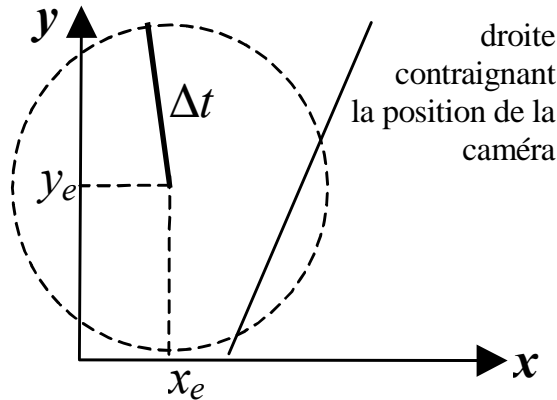
### Enoncé de la contrainte unaire

« Un appariement  $L_i-l_j$ , d'orientation correspondante  $\phi_{ij}$  et de droite de contrainte  $D_{ij}$  est cohérent si et seulement si

- $|\phi - \phi_e| < \Delta\phi$ ;

-  $D_{ij} \cap C_{ij} \neq \emptyset$ . »

$C_{ij}$  est le cercle de centre le point  $[x_e, y_e]$  et de rayon  $\Delta t_e = \sqrt{\Delta x_e^2 + \Delta y_e^2}$ .



**Figure 4.4 :** *contrainte unaire*

### Contrainte binaire

Lorsqu'un segment  $L_i$  du modèle est vertical, les paramètres  $a_{ij}$  et  $b_{ij}$  obtenus en le mettant en correspondance avec un segment  $l_j$  de la scène sont nuls. Le calcul de l'orientation ne peut pas se faire de cette façon dans ce cas là. La contrainte unaire n'est pas utilisable pour les segments verticaux du modèle. Nous allons donc définir une contrainte binaire pour ce cas. Soit un couple de segments verticaux du modèle  $(L_i, L_j)$  et un couple  $(l_k, l_l)$  de segments de l'image, supposés correspondre à  $L_i, L_j$ .

### Cas où $\theta = \pi/2$

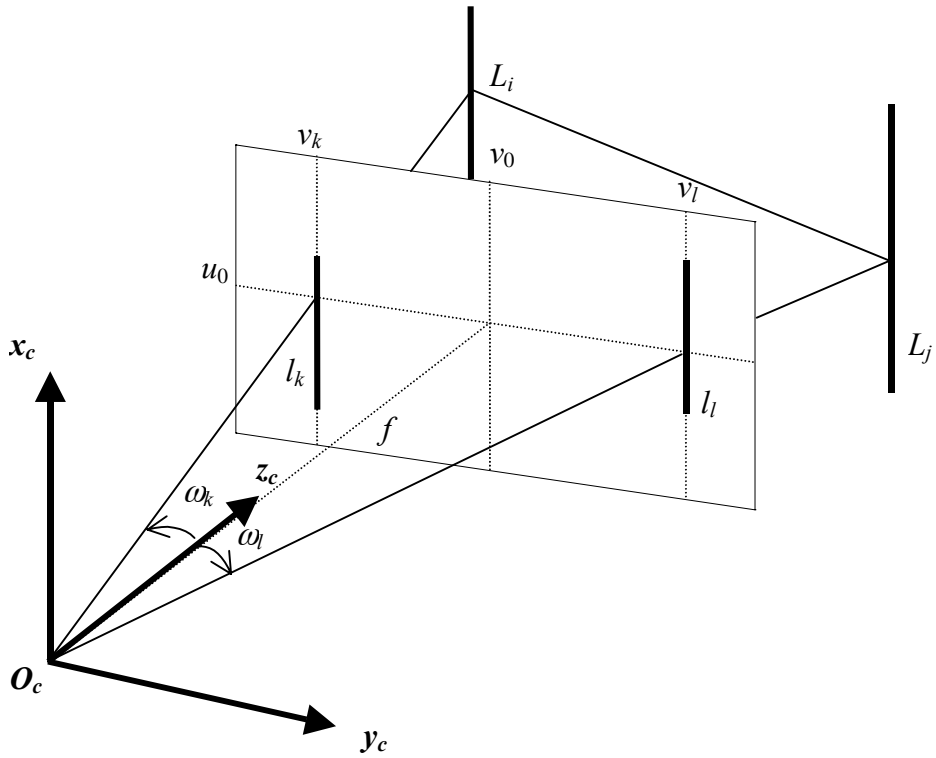
Supposons, dans un premier temps, que la caméra est dans une position horizontale, ce qui correspond à un angle  $\theta = \pi/2$ . Dans cette configuration, et vu que l'angle  $\psi$  est nul, les deux segments  $(l_k, l_l)$  de la scène sont verticaux dans le plan image. Ils peuvent être définis par les équations  $v = v_k$  et  $v = v_l$ .

A partir de la donnée  $v_k$  et connaissant les paramètres intrinsèques de la caméra, on peut calculer l'angle de vue  $\omega_k$  de chaque segment, c'est-à-dire l'angle entre le rayon optique de la caméra et la droite horizontale passant par la focale et appartenant au plan d'interprétation du segment  $l_k$  (figure 4.4).

$$\omega_k = \arctan\left(\frac{v_k - v_0}{\alpha_v}\right) \quad (4.2.4)$$

L'angle de vue entre les deux segments est ensuite calculé :

$$\omega_{kl} = |\omega_k - \omega_l| \quad (4.2.5)$$



**Figure 4.5 : contrainte binaire**

En appliquant la règle du cosinus sur le triangle formé par le centre focal et les deux points d'intersection du plan horizontal passant par ce centre et les deux segments  $L_i, L_j$  (figure 4.5) on obtient l'équation suivante :

$$d_{ij}^2 = d_i^2 + d_j^2 - 2d_i d_j \cos(\omega_{kl}) \quad (4.2.6)$$

où  $d_{ij}$  est la distance horizontale connue entre les deux segments parallèles  $L_i$  et  $L_j$  du modèle,  $d_i$  et  $d_j$  représentent respectivement les distances horizontales des segments  $L_i$  et  $L_j$  au centre focal.

$$\begin{aligned} d_i &= \sqrt{(x_i - x)^2 + (y_i - y)^2} \\ d_j &= \sqrt{(x_j - x)^2 + (y_j - y)^2} \end{aligned} \quad (4.2.7)$$

Cette équation est en fait celle d'un cercle par rapport aux coordonnées  $x$  et  $y$  du centre focal de la caméra. La position de la caméra est donc contrainte à appartenir à un arc de cercle défini par (4.2.6). De la même façon que pour la contrainte unaire, si le couple d'appariements  $(L_i-l_k, L_j-l_l)$  est correct, le cercle correspondant  $C_{ij}'$  coupe la zone d'erreur définie précédemment (figure 4.5).

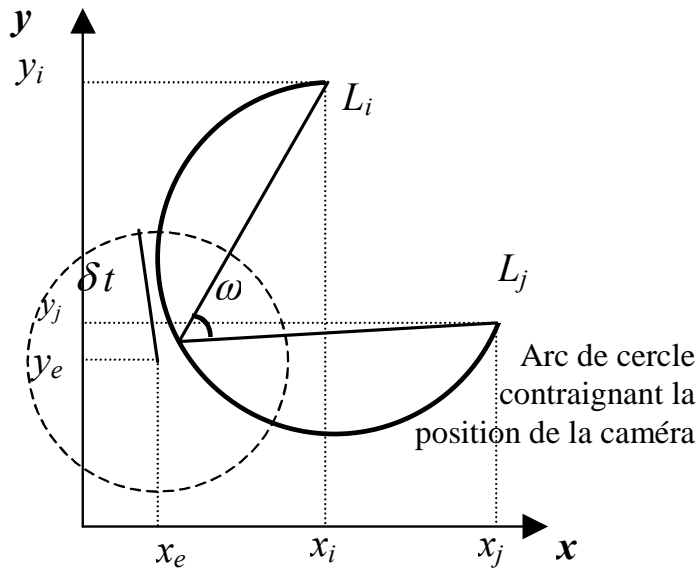


Figure 4.6 : contrainte binaire

Théoriquement, il existent deux cercles pouvant satisfaire la condition sur les angles de vue. L'un d'eux peut être éliminé en testant la cohérence dans l'ordre d'apparition de droite à gauche des deux segments sur l'image. Cet ordre reste inchangé lorsque le centre optique se déplace sur l'arc de contrainte. Il s'inverse lors du passage sur l'autre arc.

### Cas général

En pratique la caméra a un angle de site connu mais variable. Restreindre l'étude au cas où  $\theta = \pi/2$  est une contrainte importante pour l'application puisqu'elle limiterait le champ de vision de la caméra (ce qui est spécialement gênant lors d'une tâche d'exploration où la caméra est commandée par un opérateur humain). Nous allons donc généraliser le calcul précédent au cas où  $\theta \neq \pi/2$ .

Dans ce cas les projections respectives des segments  $L_i$  et  $L_j$  ne sont pas des droites verticales dans le plan image (figure 4.7). Pour pouvoir mesurer les angles de vue  $\omega$  nécessaires à l'application de notre contrainte binaire, une droite horizontale, que nous appellerons ligne d'horizon, est définie sur le plan image comme la droite de l'image dont le plan d'interprétation est horizontal et passe par le centre focal. L'équation de cette ligne d'horizon dans le plan image et la suivante :

$$u_{hor} = \tan\left(\theta - \frac{\pi}{2}\right)\alpha_u + u_0 \quad (4.2.8)$$

On démontre (annexe B) que les coordonnées  $v_i$  et  $v_j$  des intersections des droites  $l_k$ , et  $l_l$  avec la ligne d'horizon sont égales aux mesures  $v_k$  et  $v_l$  (telles que définies plus haut pour le cas où  $\theta = \pi/2$ ) que donneraient ces mêmes droites sur l'image virtuelle obtenue en posant  $\theta = \pi/2$  avec la caméra à la même position. Une fois  $v_k$  et  $v_l$  calculés on peut appliquer la contrainte binaire.

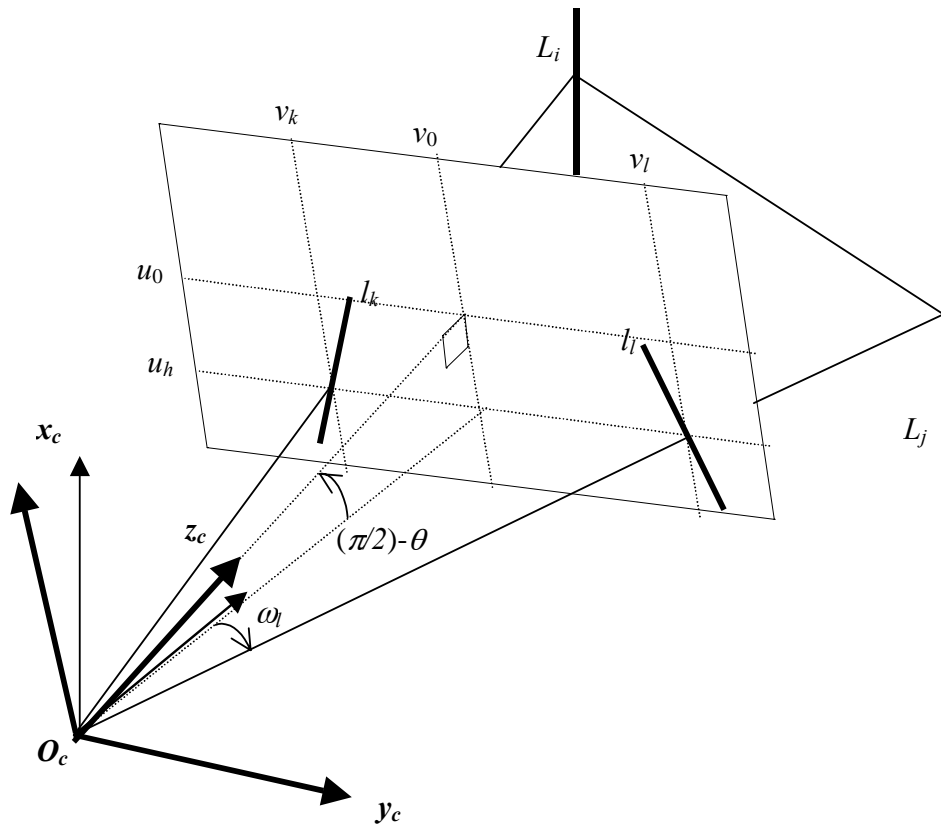


(a :  $\theta < 90^\circ$ )

(b :  $\theta = 90^\circ$ )

(c :  $\theta > 90^\circ$ )

**Figure 4.7 :** évolution de la pente du segment de projection d'un segment vertical dans le modèle en fonction de l'angle  $\theta$  de la caméra.



**Figure 4.8 :** contrainte binaire dans le cas général

### Enoncé de la contrainte binaire

« Un couple de correspondances  $(L_i-l_k, L_j-l_l)$  de cercle de contrainte  $C_{ij}$  est cohérent si et seulement si :

- $C_{ij} \cap C_{ij} \neq \emptyset$ .
- L'ordre d'apparition de droite à gauche sur l'image est cohérent. »

## Subdivision de l'arbre d'interprétation

La faiblesse principale des méthodes basées sur la recherche arborescente est l'explosion du nombre de combinaisons d'appariements. En plus des contraintes unaires et binaires nous définissons une troisième contrainte qui permet de subdiviser l'arbre d'interprétation initial en deux sous-arbres de dimensions inférieures. Ceci a pour conséquence, dans la plupart des cas, une grande réduction de l'espace des correspondances à rechercher.

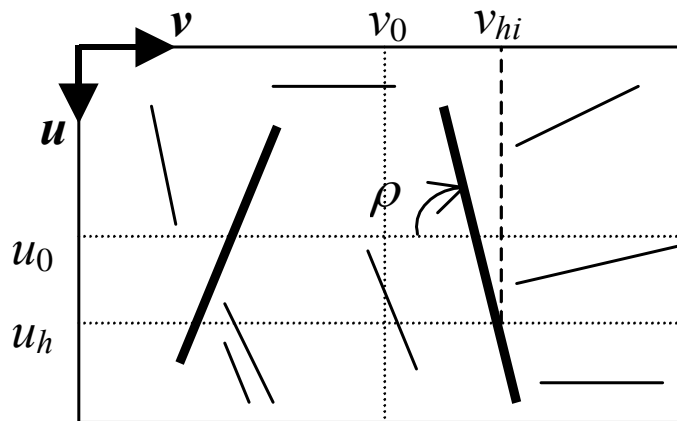
Nous avons vu dans la définition de la contrainte binaire que les projections des segments verticaux dans le modèle n'étaient pas forcément des droites verticales dans l'image.

Cependant, et compte tenu de l'intervalle des valeurs de  $\theta$  dans la plupart des applications de robots mobiles, l'orientation  $\rho$  de ces segments dans le plan image est proche de la verticale.

Il serait donc inutile de tester par exemple la cohérence de l'appariement d'un segment vertical du modèle avec une droite de l'image dont la pente est inférieure à un certain seuil (figure 4.9). Cette caractéristique peut être exploitée pour éliminer prématurément un nombre important d'hypothèses de l'espace de recherche. En effet, l'ensemble des segments de l'image correspondant potentiellement à des segments verticaux du modèle peut être défini en appliquant la règle énoncée en annexe B. En revanche, un segment du modèle qui serait horizontal ou proche de l'horizontale peu très bien se projeter sous la forme d'une droite verticale sur l'image.

En conclusion, la stratégie à adopter est la suivante :

- Subdiviser l'ensemble de segments du modèle  $\mathbf{L}=\{L_1, L_2, \dots, L_m\}$  en deux sous-ensemble  $\mathbf{L}_V=\{L_{V1}, L_{V2}, \dots, L_{Vm'}\}$  regroupant les  $m'$  segments verticaux et  $\mathbf{L}_H=\{L_{H1}, L_{H2}, \dots, L_{Hm''}\}$  regroupant les  $m''$  segments non verticaux restant du modèle.
- Extraire de  $\mathbf{l}=\{l_1, l_2, \dots, l_n\}$  un sous-ensemble  $\mathbf{l}_V=\{l_{V1}, l_{V2}, \dots, l_{Vn'}\}$  regroupant les droites images dont la pente satisfait la règle définie en annexe B.
- Construire un arbre d'interprétation avec les ensembles  $\mathbf{L}_V$  et  $\mathbf{l}_V$  puis les élaguer grâce à la contrainte binaire. Le nombre maximal de combinaisons sur lesquelles peut être appliquée cette contrainte dans cet arbre est obtenu en multipliant les  $C_2^{n'}$  arrangements possibles de deux segments image par les  $C_2^{m'}$  combinaisons possible de deux segments du modèle soit  $N_v = C_2^{m'} \cdot C_2^{n'}$ .
- Construire un arbre d'interprétation avec les ensembles  $\mathbf{L}_H$  et  $\mathbf{l}$  puis les élaguer grâce à la contrainte unaire. Le nombre maximal de combinaisons à explorer est  $N_h = m'' \cdot n$ .



*Figure 4.9 : seules les lignes en gras peuvent correspondre à des segments verticaux*

### Génération des hypothèses de correspondance

A l'issue de l'élagage des deux arbres d'interprétation définis ci-dessus, nous obtenons deux listes d'appariements possibles. La liste d'appariements d'une primitive de  $\mathbf{L}_H$  avec une primitive de  $\mathbf{l}$  obtenue grâce à la contrainte unaire est utilisée pour constituer les interprétations partielles de la scène en combinant ces appariements.

De la même façon, les interprétations partielles possibles sont établies en combinant les appariements de couples de  $\mathbf{L}_H$  et de  $\mathbf{l}_v$  retenus après application de la contrainte binaire. Un dernier contrôle est effectué en testant l'ordre d'apparition de droite à gauche des segments sur l'image et en le comparant avec celui attendu. En effet, un ordre d'apparition peut être établi sur les segments de la liste de visibilité de chaque RIV connaissant une estimation de l'orientation. Le calcul est détaillé en Annexe C.

Les deux listes d'interprétations partielles sont elles mêmes combinées pour constituer une dernière liste d'interprétations globales possibles de la scène ou hypothèses de mise en correspondances. Le nombre d'hypothèses ainsi retenues est donc égal au produit des effectifs des deux listes d'hypothèses.

### Vérification des hypothèses

L'application des contraintes de bas niveau (unaire et binaire) permet de vérifier la cohérence locale d'appariements de primitives. Ces contraintes servent à éliminer les fausses correspondances dont l'incohérence est évidente. Il est important de réaliser que l'application des contraintes précédemment définies ne va pas rejeter la totalité des fausses hypothèses de correspondances. Une interprétation composée à partir d'appariements localement cohérents n'est pas forcément globalement cohérente. Mais toutes celles retenues restent potentiellement correctes. Une vérification plus fine doit leur être appliquée.

Puisque les interprétations globales sont composées par les combinaisons des appariements ayant passé l'étape d'élagage uniquement, nous disposons, à ce stade, d'un nombre beaucoup plus restreint d'hypothèses à vérifier. On peut se à présent permettre une méthode plus coûteuse en temps de calcul comme le calcul de la transformation entre le repère caméra et le repère du modèle. La méthode adoptée consiste, pour chaque interprétation ou hypothèse de mise en correspondance, à calculer la transformation correspondant puis à projeter les segments de  $\mathbf{L}$  sur l'image. Le problème est alors de définir une fonction permettant de donner une estimation quantitative du degré de ressemblance entre l'image réelle et l'image virtuelle obtenue par projection.

L'utilisation de la fonction d'erreur minimisée pour le calcul de la position et de l'orientation de la caméra comme cela est le cas dans ([Chang99]) ne serait pas une bonne idée car elle est basée sur l'orientation des segments. Il faut définir une fonction qui tient à la fois compte de l'orientation mais aussi de la position des segments dans l'image. C'est le propos des sections suivantes.

### Distance entre deux segments de l'image

La métrique généralement utilisée pour exprimer la distance entre deux segments de droites de même longueur liés par un déplacement rigide est obtenue en calculant la somme des carrés des distances entre les points correspondants sur chacun des segments ([Kamgar-Parsi97]). Par correspondant on entend les points situés à égales distances de l'extrémité de chaque segment. Soient deux segments  $s_1, s_2$  de longueur  $\lambda$  dont les points milieux  $m_1, m_2$  ont respectivement comme vecteurs de position  $\mathbf{m}_1, \mathbf{m}_2$  et comme vecteurs directeurs  $\mathbf{v}_1, \mathbf{v}_2$ . La distance euclidienne entre deux points situés à une distance  $\alpha$  de  $\mathbf{m}_1, \mathbf{m}_2$  est  $D(\alpha)$  telle que

$$D^2(\alpha) = \|(\mathbf{m}_1 - \mathbf{m}_2) + \alpha(\mathbf{v}_1 - \mathbf{v}_2)\| \quad (4.2.9)$$

La distance  $D(s_1, s_2)$  entre les deux segments peut alors s'exprimer par :

$$D(s_1, s_2) = \int_{-\frac{\lambda}{2}}^{\frac{\lambda}{2}} D^2(\alpha) d\alpha = \lambda \|\mathbf{m}_1 - \mathbf{m}_2\|^2 + \frac{\lambda^3}{6} (1 - \mathbf{v}_1^T \mathbf{v}_2) \quad (4.2.10)$$

Comme on peut s'y attendre, cette distance dépend de la séparation de leurs points milieux et de leurs orientations relatives.

Soient, maintenant, deux segments de droites  $l_p$ , d'extrémités  $m_p^s$  et  $m_p^e$  obtenues par projection d'un segment  $L$  du modèle sur le plan image et  $l$ , d'extrémités  $m^s$  et  $m^e$  son segment correspondant observé sur l'image.  $l_p$  et  $l$  sont tous deux définis dans un repère 2D lié au plan image  $(O, \mathbf{u}, \mathbf{v})$ . En raison des occultations, la longueur de  $l$  est souvent inférieure à celle de  $l_p$ . Par ailleurs, la transformation entre les deux segments n'est pas rigide. La métrique ci-dessus doit être adaptée. Nous exprimons la distance entre les deux segments  $l_p$  et  $l$  par la somme des distances euclidiennes entre chaque point de  $l$  et son plus proche point sur  $l_p$ . Pour calculer cette distance nous définissons un repère 2D  $(m_p^s, \mathbf{x}, \mathbf{y})$ ,

$$\text{avec } \mathbf{x}_p = \frac{\overrightarrow{m_p^s m_p^e}}{\|m_p^s m_p^e\|}, \quad \mathbf{y}_p = \frac{\overrightarrow{m_p^s m_i}}{\|m_p^s m_i\|}$$

et  $m_i$  le point d'intersection de la droite porteuse du segment  $l$  et de la perpendiculaire à  $l_p$  passant par  $m_p^s$  (figure 4.10).

Soit  $y = ax + b$  l'équation de la droite porteuse du segment  $l$  dans ce repère.

Pour un point quelconque  $m$  de  $l$  de coordonnées  $(x_m, y_m)$ , le plus proche point sur  $l_p$  est :

- $m_p^s$  si  $x_m \leq 0$ ,
- sa projection perpendiculaire sur  $l_p$  si  $0 < x_m < \lambda_p$  ( $\lambda_p$  étant la longueur de  $l_p$ ),
- $m_p^e$  si  $x_m \geq \lambda_p$ .

La distance entre les deux points s'écrit donc

$$- d(x_m) = \sqrt{x_m^2 + (y_m - y_p^s)^2} \quad \text{si } x_m^s \leq 0,$$



- $d(x_m) = (\text{distance perpendiculaire à } l_p)$  si  $0 < x_m < \lambda_p$  ( $\lambda_p$  étant la longueur de  $l_p$ ),
- $d(x_m) = \sqrt{(x_m - x_p^e)^2 + (y_m - y_p^e)^2}$  si  $x_m \geq \lambda_p$ .

Nous obtenons donc l'expression de la distance entre deux segments dans un plan

$$D(l_p, l) = \frac{1}{\lambda_p} \int_{x^s}^{x^e} d(x) dx \quad (4.2.11)$$

Le terme  $1/\lambda_p$  permet de normaliser la mesure pour ne pas pénaliser les appariements mettant en jeu des segments longs.

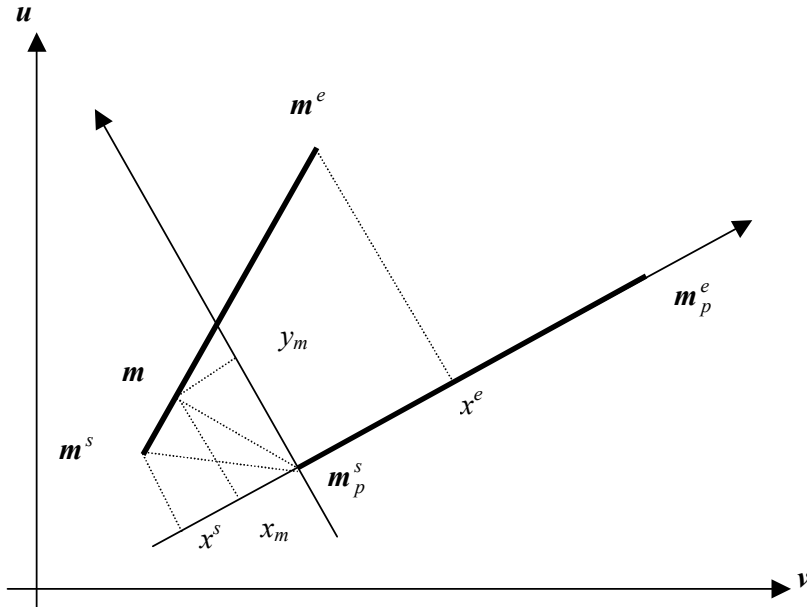


Figure 4.10 : notion de distance entre deux segments dans le plan image

### Distance entre deux ensembles de segments

Nous allons à présent, utiliser la fonction de distance entre deux segments définie plus haut pour déduire une fonction de « score » pour chaque hypothèse de correspondance entre l'ensemble des segments du modèle et ceux de l'image. Autrement dit, définir une distance entre les deux ensembles de segments que sont l'ensemble de segments observés sur l'image et l'ensemble des segments virtuels issus de la projection des segments du modèle partiel sur le plan image grâce à la transformation calculée en utilisant l'hypothèse de correspondance. La façon la plus simple est de calculer la somme des distances des appariements composant l'hypothèse de correspondance. Le problème est que la fonction ainsi obtenue tendrait à favoriser les hypothèses de correspondance avec un faible nombre d'appariements.

La solution que nous proposons est inspirée de la notion de distance de Hausdorff qui est une mesure mathématique du degré de ressemblance entre deux ensembles de primitives géométriques. La formulation la plus connue est donnée pour deux ensembles de points comme suit :

#### Définition

Etant donné deux ensembles de points  $A = \{a_1, a_2, \dots, a_m\}$  et  $B = \{b_1, b_2, \dots, b_n\}$ , la distance de Hausdorff  $H(A, B)$  entre  $A$  et  $B$  est

$$H(A, B) = \max(h(A, B), h(B, A)) \quad (4.2.12)$$

Où  $h(A,B)$  est la distance de hausdorff unidirectionnelle définie par

$$h(A, B) = \max_{a_i \in A} (\min_{b_j \in B} d(a_i, b_j)) \quad (4.2.13)$$

avec  $d$  la distance euclidienne entre  $a_i$  et  $b_j$ .

Partant du même principe, la fonction de score pour une hypothèse de correspondance est définie comme suit :

**Définition**

Etant donnés deux ensembles de segments,  $L_p = \{l_{p1}, l_{p2}, \dots, l_{pm}\}$  et  $L = \{l_1, l_2, \dots, l_m\}$  respectivement appariés, la distance entre ces deux ensembles est

$$h(L_p, L) = \max_{i=1, \dots, m} (D(l_{pi}, l_i)) \quad (4.2.14)$$

Les hypothèses de correspondance ne sont pas composées du même nombre d'appariements. Aussi, nous choisissons d'appliquer la fonction précédente sur un nombre fixe  $k$  des segments les mieux appariés.  $k$  est choisi comme étant le nombre minimum d'appariements nécessaires pour valider une hypothèse de correspondance. Il existe plusieurs travaux sur la façon de déterminer la valeur de  $k$  ([Pennec98]). Dans ce qui suit nous considérons  $k$  connu. La fonction de distance s'écrit alors

$$h(L_p, L) = \max_{i=1, \dots, k} (k \min(D(l_{pi}, l_i)) \quad (4.2.15)$$

où  $k$ min représente les  $k$  segments  $l_i$  les plus proches de leurs correspondants dans le sens de la distance  $D$  définie plus haut.

**Récapitulatif de l'algorithme**

La figure 4.11 montre un schéma synoptique qui résume la stratégie de recherche de la meilleure hypothèse de mise en correspondance.

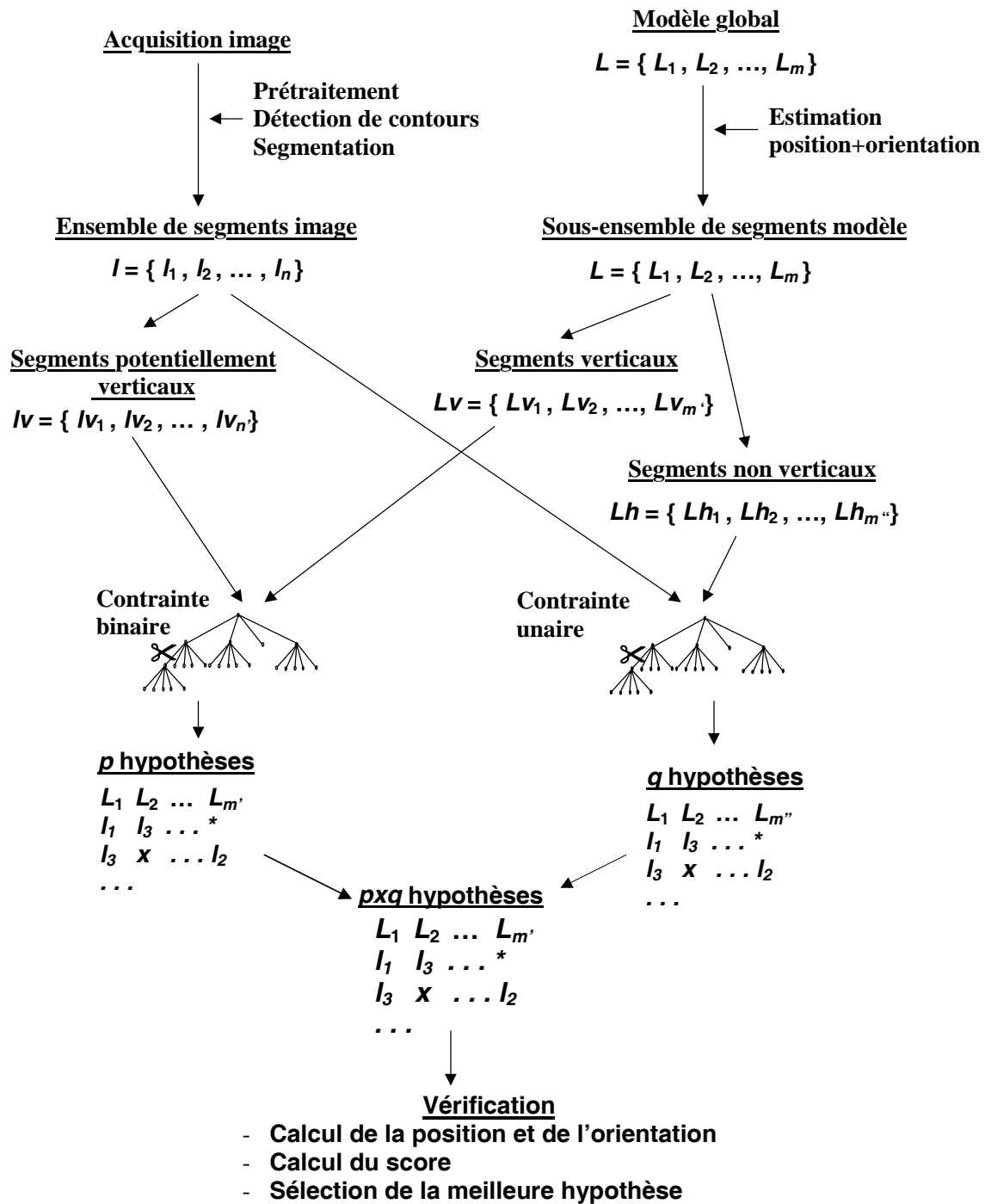


Figure 4.11 : schéma synoptique de l'algorithme de mise en correspondance

## Prise en compte de l'erreur

Les contraintes unaire et binaire utilisées pour écarter les appariements non cohérents sont basées sur la marge d'erreur de la localisation initiale mais aussi sur les valeurs des paramètres des segments observés sur l'image. Comme nous l'avons vu, ces valeurs sont soumises à plusieurs sources de bruit. Il en résulte que l'incertitude sur les paramètres  $\rho$  et  $d$  du segment de l'image utilisé pour un appariement se propage sur les paramètres des zones de contraintes de l'orientation et de la position (droite pour la contrainte unaire et arc de cercle pour la contrainte binaire).

Pour la contrainte unaire, la variation du vecteur de paramètres  $[A, B, C]^T$  de la droite de contrainte obtenue dans un domaine défini transforme la zone de contrainte en une région du plan de déplacement du robot délimitée par l'enveloppe de l'ensemble des droites définies par les différentes valeurs du vecteur  $[A, B, C]^T$  (figure 4.12). De même que la variation du vecteur  $[c, r]^T$  (centre et rayon) définissant l'arc de cercle obtenu grâce à l'application de la contrainte binaire produit une zone de contrainte schématisée par la figure 4.13.

Pour éviter d'éliminer des appariements corrects, il faut rendre la sélection grâce aux contraintes géométriques plus souples en introduisant des seuils sur les valeurs des paramètres de contrainte. Cependant, la détermination de la valeur de ces seuils est déterminante. Des seuils trop sélectifs risqueraient de faire rater de bons appariements. Inversement des seuils surestimés provoqueraient l'acceptation d'un trop grand nombre d'appariements réduisant l'efficacité de l'étape d'élagage et augmentant ainsi la probabilité d'une fausse mise en correspondance.

Dans ce qui suit nous allons présenter deux approches pour la propagation de l'erreur sur les paramètres des segments observés sur l'image aux paramètres de droites et cercles des contraintes unaire et binaire. Un vecteur de paramètres définissant un segment observé est considéré comme un événement aléatoire appartenant à une distribution gaussienne. Dans la première approche, les seuils de variation des paramètres des zones de contraintes sont déterminés empiriquement et restent fixes. Dans la deuxième approche les seuils sont déterminés par propagation des matrices de covariance des paramètres d'entrée.

## Détermination des seuils par une approche empirique

Cette approche consiste à utiliser une méthode de Monté-Carlo ([Neal93]) pour estimer la moyenne et les écarts types des paramètres des zones de contrainte lorsque les paramètres du segment de l'image varient suivant une distribution Gaussienne.

Pour déterminer les seuils concernant la zone de la contrainte unaire, un ensemble d'appariements segment modèle - segment image représenté par son vecteur de paramètres  $[\rho, d]^T$  est généré de façon aléatoire. Un bruit de distribution uniforme est ensuite appliqué à  $\rho$  et  $d$  puis le vecteur de paramètres de la droite de contrainte est recalculé. Un nombre significatif d'échantillons sont ainsi générés pour chaque appariement. La valeur moyenne et la matrice de covariance des paramètres de sortie obtenus sont calculées. Le seuil est défini par rapport à ces écarts types obtenus à partir de la diagonale de la matrice de covariance. Un raisonnement similaire est effectué avec des couples d'appariements pour déterminer les seuils de variation des paramètres  $[c, r]^T$  des arcs de cercle de la contrainte binaire. Cette approche a le mérite de réduire les temps de calcul puisque celui-ci est effectué hors-ligne. Cependant, les seuils ainsi obtenus sont supposés fixes pour tout appariement possible. Ceci nous amène à majorer leur valeur pour garantir de ne pas éliminer des appariements corrects en raison du bruit. Une autre façon de procéder, est de calculer en ligne une estimation de ces seuils pour chaque appariement. C'est le calcul développé dans la section suivante.

## Détermination des seuils par une approche probabiliste

### Rappels mathématiques

#### propagation des matrices de covariance par la Jacobienne

Soit une variable aléatoire de dimension quelconque, définie par le vecteur  $\mathbf{x}$ , de valeur moyenne  $\bar{\mathbf{x}}$  et de matrice de covariance  $\Sigma_{\mathbf{x}}$ .

$$\bar{\mathbf{x}} = E(\mathbf{x}) \quad (4.2.16)$$

$$\Sigma_{\mathbf{x}} = E((\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T) \quad (4.2.17)$$

Si  $\mathbf{y}$  est la transformée de  $\mathbf{x}$  grâce à une fonction  $f$ , alors la valeur moyenne et la matrice de covariance de  $\mathbf{y}$  sont données par :

$$\bar{\mathbf{y}} = f(\bar{\mathbf{x}}) \quad (4.2.18)$$

$$\Sigma_{\mathbf{y}} \approx \mathfrak{J}_f(\bar{\mathbf{x}}) \cdot \Sigma_{\mathbf{x}} \cdot \mathfrak{J}_f^T(\bar{\mathbf{x}}) \quad (4.2.19)$$

où  $\mathfrak{J}_f$  est la matrice Jacobienne de  $f$ .

#### Distance de Mahalanobis

Grâce aux équations précédentes nous pouvons propager les variations du vecteur aléatoire  $\mathbf{x}$  sur  $\mathbf{y}$ . Il s'agit maintenant de borner les variations de  $\mathbf{y}$  connaissant les limites de variation de  $\mathbf{x}$ . Par définition de la distribution Gaussienne, Il existera toujours une probabilité non nulle d'occurrence d'un événement  $\mathbf{x}$  quelque soit l'endroit où l'on se place dans l'espace de définition de  $\mathbf{x}$ . Ceci est catastrophique pour notre application car impliquant que toutes les hypothèses d'appariement doivent être validées puisque la surface de contrainte de la position correspondrait à chaque fois à tout l'espace de travail.

Pour éviter ce problème nous utilisons la distance de Mahalanobis pour définir une zone de l'espace de définition de la variable gaussienne  $\mathbf{x}$  dans laquelle la probabilité pour que cette variable ait une valeur est supérieure à un seuil déterminé. Cette zone est définie par

$$H_{\mathbf{x}}(d) = \left\{ \mathbf{x} \in \mathbb{R}^n / d_{\bar{\mathbf{x}}, \Sigma_{\mathbf{x}}}(\mathbf{x}) \leq d \right\} \quad (4.2.20)$$

avec  $d_{\bar{\mathbf{x}}, \Sigma_{\mathbf{x}}}(\mathbf{x})$  distance de Mahalanobis définie par

$$d_{\bar{\mathbf{x}}, \Sigma_{\mathbf{x}}}(\mathbf{x}) = \sqrt{(\mathbf{x} - \bar{\mathbf{x}})^T \Sigma_{\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}})} \quad (4.2.21)$$

La valeur de  $d$  est liée à la probabilité d'occurrence de l'événement  $\mathbf{x}$ . Cette probabilité est  $2I(\mathbf{x})-1$  ou  $I(\mathbf{x})$  est la distribution gaussienne. Par exemple, pour  $d$  égal à 2 la probabilité est de 95 %. La région  $H_{\mathbf{x}}(d)$  ainsi définie par le choix d'une valeur appropriée de  $d$  est en fait un hyper ellipsoïde dans l'espace de paramétrisation de la variable  $\mathbf{x}$ .

Faugeras ([Faugeras93]) montre que le mécanisme de propagation des covariances de  $\mathbf{x}$  peut s'interpréter de façon déterministe comme la propagation d'ordre un de la zone d'incertitude de  $\mathbf{y}$ . En d'autres termes, si  $f$  est une fonction continue et dérivable et  $Z_d(\mathbf{x})$  une zone d'incertitude de la variable  $\mathbf{x}$ , alors la zone d'incertitude obtenue  $Z_d(f(\mathbf{x}))$  est une approximation d'ordre un de la zone  $f(Z_d(\mathbf{x}))$ .

#### Prise en compte de l'erreur dans la contrainte unaire

Considérons un appariement d'une droite de l'image avec un segment du modèle. Lorsque les paramètres de la droite de l'image varient, l'orientation calculée et les paramètres de la droite de contrainte varient également dans un certain domaine autour de leur valeur moyenne, l'ensemble de droites de contrainte sur la position ainsi obtenu définit une surface dans le plan

de déplacement de la caméra (figure 4.12) Pour délimiter le domaine de variation de l'orientation et la surface de contrainte sur la position de la caméra, obtenus par l'appariement d'un segment  $L_i$  du modèle avec un segment  $l_j$  de l'image, il faut propager l'incertitude sur les paramètres  $\rho_j$  et  $d_j$  du segment  $l_j$  à  $\phi_j$  et aux paramètres  $A_{ij}$ ,  $B_{ij}$  et  $C_{ij}$  de la droite de contrainte. Pour ce faire, nous allons d'abord exprimer  $\phi_j$ ,  $A_{ij}$ ,  $B_{ij}$  et  $C_{ij}$  en fonction de  $\rho_j$  et  $d_j$ . On cherche donc deux fonctions  $f_\phi$  et  $f_{ABC}$  telles que :

$$\begin{cases} \phi = f_\phi([\rho_{ij}, d_{ij}]^T) \end{cases} \quad (4.2.22)$$

$$\begin{cases} [A_{ij}, B_{ij}, C_{ij}]^T = f_{ABC}([\rho_{ij}, d_{ij}]^T) \end{cases} \quad (4.2.23)$$

On démontre en Annexe A que le vecteur normal au plan d'interprétation d'un segment  $l_j$  de l'image peut s'écrire en fonction de  $\rho_j$  et  $d_j$  comme suit :

$$\begin{cases} n_x = \cos(\rho_j) \\ n_y = -\sin(\rho_j) \\ n_z = \frac{1}{\alpha}(d_j - u_0 \cos(\rho_j) - v_0 \sin(\rho_j)) \end{cases} \quad (4.2.24)$$

La solution de l'équation (4.2.1) donne l'expression de l'orientation (définissant la fonction  $f_\phi$ ) :

$$\phi = \arctan\left(\frac{b_{ij}}{a_{ij}}\right) \pm \arccos\left(\frac{c_{ij}}{\sqrt{a_{ij}^2 + b_{ij}^2}}\right) \quad (4.2.25)$$

où  $a_{ij}$ ,  $b_{ij}$  et  $c_{ij}$  sont les mêmes que dans l'équation (4.2.1), exprimés en remplaçant les composantes du vecteur normal par les expressions de (4.2.24).

De la même façon on obtient les paramètres de la droite de contrainte  $A_{ij}$ ,  $B_{ij}$  et  $C_{ij}$  comme suit (c'est la définition de la fonction  $f_{ABC}$ ) :

$$\begin{cases} A_{ij} = A_{1i}n_x \cos(\phi) + A_{2i}n_y \sin(\phi) + A_{3i}n_z \cos(\phi) \\ B_{ij} = B_{1i}n_x \sin(\phi) + B_{2i}n_y \cos(\phi) + B_{3i}n_z \sin(\phi) \\ C_{ij} = [C_{1i}n_x + C_{2i}n_y + C_{3i}n_z] \cos(\phi) + [C_{4i}n_x + C_{5i}n_y + C_{6i}n_z] \sin(\phi) + [C_{7i}n_x + C_{8i}n_y + C_{9i}n_z] \end{cases} \quad (4.2.26)$$

avec comme valeurs des constantes :

$$\begin{cases} \begin{cases} A_{1i} = -\cos(\theta) \\ A_{2i} = -1 \\ A_{3i} = \sin(\theta) \end{cases} & \begin{cases} B_{1i} = \cos(\theta) \\ B_{2i} = -1 \\ B_{3i} = -\sin(\theta) \end{cases} & \begin{cases} C_{1i} = x_i \cos(\theta) \\ C_{2i} = y_i \\ C_{3i} = x_i \sin(\theta) \\ C_{4i} = -y_i \cos(\theta) \\ C_{5i} = x_i \\ C_{6i} = y_i \sin(\theta) \\ C_{7i} = (z_i - z) \sin(\theta) \\ C_{8i} = 0 \\ C_{9i} = (z_i - z) \cos(\theta) \end{cases} \end{cases}$$

En supposant que  $[\rho_j, d_j]^T$  est un vecteur aléatoire appartenant à une distribution gaussienne de valeur moyenne  $[\bar{\rho}_j, \bar{d}_j]^T$  et de matrice de covariance  $\Sigma_{\rho d}$ , représentant les variations des paramètres du segment de l'image par effet du bruit.

La matrice de covariance  $\Sigma_{\rho d}$  s'écrit sous la forme :

$$\Sigma_{\rho d} = \begin{bmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_d^2 \end{bmatrix} \quad (4.2.27)$$

où  $\sigma_\rho$  et  $\sigma_d$  sont les écarts types de  $\rho$  et  $d$  déterminés empiriquement. La valeur moyenne considérée est le vecteur de paramètres du segment observé sur l'image.

Les valeurs moyennes  $[\bar{A}_{ij}, \bar{B}_{ij}, \bar{C}_{ij}]^T$  et  $\bar{\phi}$  et les matrices de covariance  $\Sigma_{YABC}$  et  $\Sigma_\phi$  s'écrivent :

$$\begin{cases} \bar{\phi} = f_\phi([\bar{\rho}_{ij}, \bar{d}_{ij}]^T) & (4.2.28) \\ [\bar{A}_{ij}, \bar{B}_{ij}, \bar{C}_{ij}]^T = f_{ABC}([\bar{\rho}_{ij}, \bar{d}_{ij}]^T) & (4.2.29) \end{cases}$$

et

$$\begin{cases} \Sigma_\phi \approx \mathfrak{S}_{f_\phi}(\bar{\rho}_{ij}, \bar{d}_{ij}) \cdot \Sigma_{\rho d} \cdot \mathfrak{S}_{f_\phi}^T(\bar{\rho}_{ij}, \bar{d}_{ij}) & (4.2.30) \\ \Sigma_{ABC} \approx \mathfrak{S}_{f_{ABC}}(\bar{\rho}_{ij}, \bar{d}_{ij}) \cdot \Sigma_{ABC} \cdot \mathfrak{S}_{f_{ABC}}^T(\bar{\rho}_{ij}, \bar{d}_{ij}) & (4.2.31) \end{cases}$$

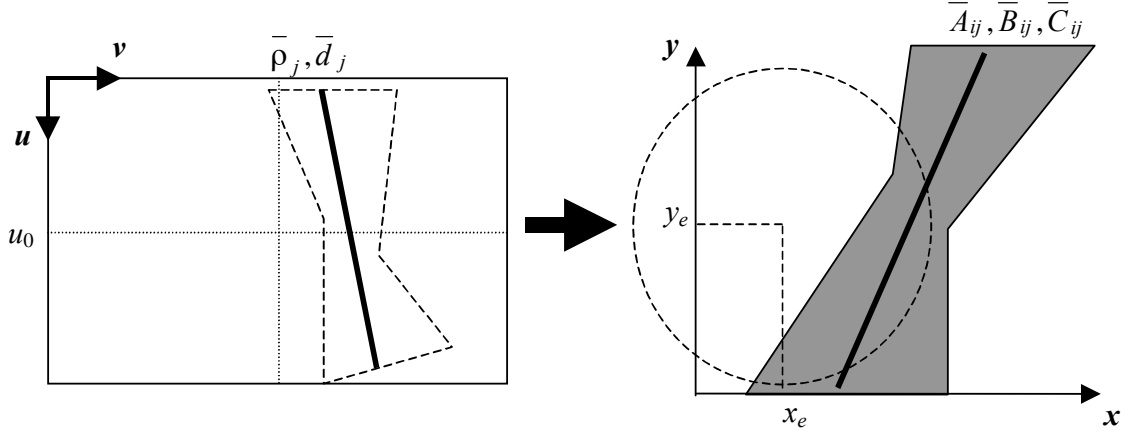
où  $\mathfrak{S}_{f_\phi}$  et  $\mathfrak{S}_{f_{ABC}}$  sont les Jacobiennes respectives de  $f_\phi$  et  $f_{\phi ABC}$  :

$$\mathfrak{S}_{f_{ABC}} = \begin{bmatrix} \frac{\partial A_{ij}}{\partial \rho_j} & \frac{\partial A_{ij}}{\partial d_j} \\ \frac{\partial B_{ij}}{\partial \rho_j} & \frac{\partial B_{ij}}{\partial d_j} \\ \frac{\partial C_{ij}}{\partial \rho_j} & \frac{\partial C_{ij}}{\partial d_j} \end{bmatrix} \quad (4.2.32) \quad \text{et} \quad \mathfrak{S}_{f_\phi} = \begin{bmatrix} \frac{\partial f_\phi}{\partial \rho_j} & \frac{\partial f_\phi}{\partial d_j} \end{bmatrix} \quad (4.2.33)$$

Le calcul détaillé de ces matrices est donné en annexe A.

Une fois les matrices de covariances calculées, les écarts types, obtenus sur les diagonales de ces matrices servent à délimiter le domaine autour de  $\bar{\phi}$  et les zones de l'espace paramétrique  $(A_{ij}, B_{ij}, C_{ij})$  autour de  $(\bar{A}_{ij}, \bar{B}_{ij}, \bar{C}_{ij})$  dans lesquels doivent se situer les réponses à la contrainte unaire. Si le domaine de  $\phi$  coupe l'intervalle d'incertitude de l'estimation initiale et s'il existe au moins une droite de contrainte, dans la zone de variation déterminée, qui coupe le cercle d'incertitude de l'estimation de la position, alors l'appariement  $L_i-l_j$  est retenu. Sinon il est rejeté.

Si l'on utilise la distance de Mahalanobis, alors on cherche l'existence d'une orientation et d'une droite de contrainte ayant une distance inférieure au seuil et coupant les domaines d'incertitude de l'estimation initiale. Cette option permet de fixer un seuil probabiliste fixe plutôt qu'un seuil métrique grâce aux écarts types.



**Figure 4.12 :** propagation de l'incertitude des paramètres des primitives de l'image sur la zone de contrainte de la position avec la contrainte uniaxe.

### Prise en compte de l'erreur dans la contrainte binaire

De la même façon que pour la contrainte uniaxe, nous allons propager l'incertitude des paramètres  $v_k$  et  $v_l$  d'un couple de segments image  $l_k$  et  $l_l$ , appariés respectivement avec les segments verticaux  $L_i$  et  $L_j$  du modèle, sur les paramètres  $r_{kl}$ ,  $x_{kl}$  et  $y_{kl}$  du cercle de contrainte obtenu par application de la contrainte binaire ( $r_{kl}$ ,  $x_{kl}$  et  $y_{kl}$  étant respectivement le rayon et les coordonnées du centre de ce cercle). L'ensemble de cercles obtenus par variation de ces paramètres définit la surface de contrainte de la position du robot (figure 4.13).

On suppose donc une distribution Gaussienne du vecteur de mesures  $[v_k, v_l]^T$  de moyenne  $[\bar{v}_k, \bar{v}_l]^T$  (égale à la valeur observée sur l'image) et de matrice de covariance  $\Sigma_{vv}$  (déterminée empiriquement).

La fonction  $f_{rxy}$  liant les deux vecteurs de paramètres

$$\begin{bmatrix} r_{kl} & x_{kl} & y_{kl} \end{bmatrix}^T = f_{rxy} \left( \begin{bmatrix} v_k & v_l \end{bmatrix}^T \right) \quad (4.2.34)$$

On cherche à déterminer la moyenne  $\begin{bmatrix} \bar{r}_{kl} & \bar{x}_{kl} & \bar{y}_{kl} \end{bmatrix}^T$  et la matrice de covariance  $\Sigma_{rxy}$  de la même que dans la section précédente.

Considérons comme position de caméra un point du cercle  $p(x,y)$  tel que le triangle formé par  $p$ ,  $p_i(x_i, y_i)$  et  $p_j(x_j, y_j)$  (les deux derniers points étant les intersections des segments du modèle  $L_i$  et  $L_j$  avec le plan horizontal) soit un triangle rectangle (figure 4.14). Dans cette configuration nous avons :

$$r_{kl} = \left\| \overrightarrow{p_i p_j} \right\| \frac{1}{2 \sin(\omega_{kl})} \quad (4.2.35)$$

avec

$$\omega_k = \arctan\left(\frac{v_k - v_0}{\alpha_v}\right) - \arctan\left(\frac{v_l - v_0}{\alpha_v}\right) \quad (4.2.36)$$

La relation de pythagore dans le triangle rectangle et l'expression de la distance du point  $p$  à la droite  $(p_i p_j)$  donne le système d'équations suivant :

$$\begin{cases} (x - x_j)^2 + (y - y_j)^2 - 4r_{kl}^2 = 0 & (4.2.37) \\ \cos(\rho_{ij})x + \sin(\rho_{ij})y - d_{ij} - r_{kl} \cos(\omega_{kl}) = 0 & (4.2.38) \end{cases}$$



et  $\rho_{ij}$  et  $d_{ij}$  les paramètres de la droite ( $p_i p_j$ ).

Après calcul nous obtenons une expression de la forme :

$$x = ar_{kl} \cos(\omega_{kl}) + b \pm \sqrt{c(r_{kl} \cos(\omega_{kl}))^2 + dr_{kl} \cos(\omega_{kl}) + e} \quad (4.2.39)$$

où  $a, b, c, d$  et  $e$  sont des constantes.

Une fois  $x$  et  $y$  calculés nous obtenons les coordonnées de  $x_c$  et  $y_c$  du centre du cercle de contrainte par

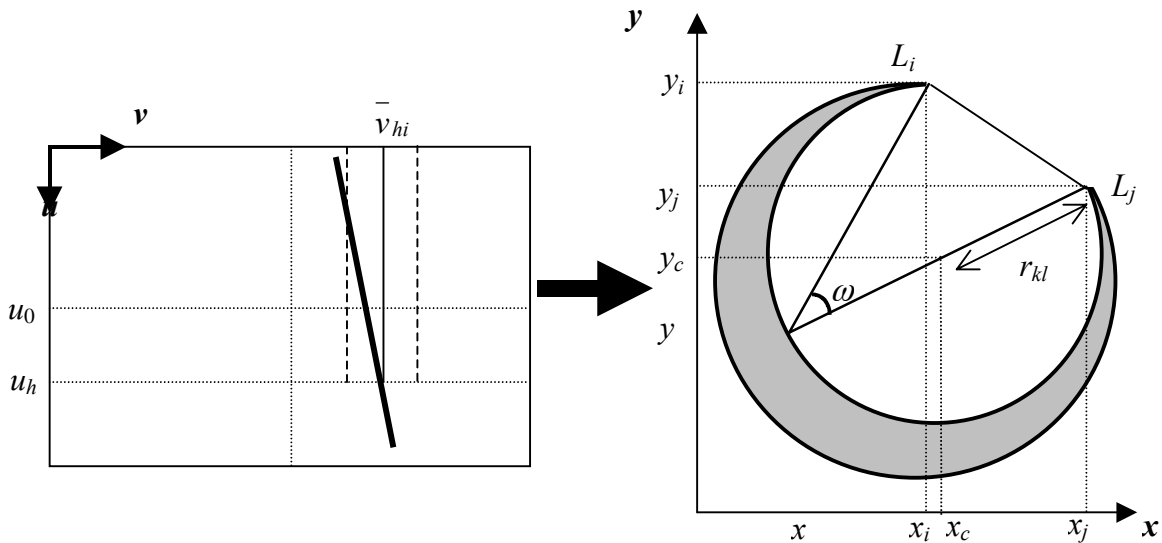
$$\begin{cases} x_c = \frac{x + x_i}{2} & (4.2.40) \\ y_c = \frac{y + y_i}{2} & (4.2.41) \end{cases}$$

La matrice Jacobienne  $\mathcal{J}_{f_{rxy}}$  recherchée s'écrit :

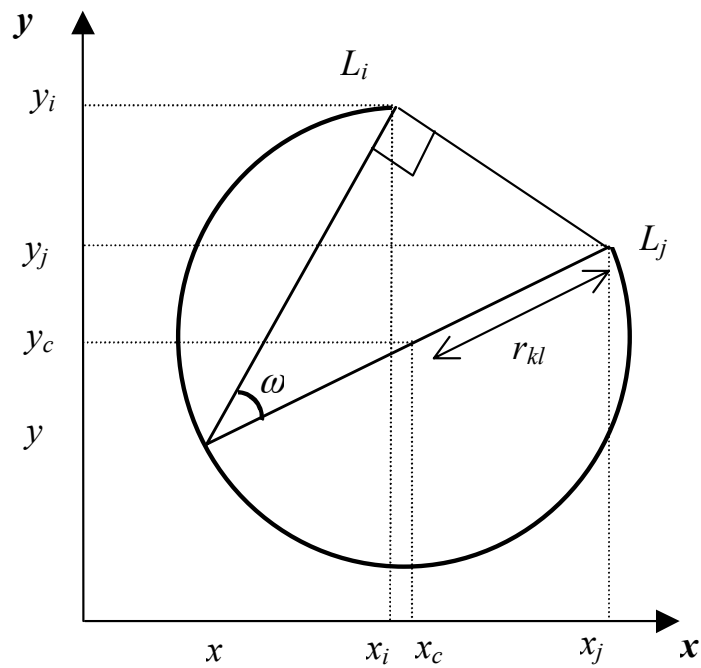
$$\mathcal{J}_{f_{rxy}} = \begin{bmatrix} \frac{\partial r_{kl}}{\partial v_k} & \frac{\partial r_{kl}}{\partial v_l} \\ \frac{\partial x_{kl}}{\partial v_k} & \frac{\partial x_{kl}}{\partial v_l} \\ \frac{\partial y_{kl}}{\partial v_k} & \frac{\partial y_{kl}}{\partial v_l} \end{bmatrix} \quad (4.2.42)$$

Le calcul des dérivées partielles est donné en Annexe A.

Au final, l'appariement d'un couple de  $L_i-L_j$  de segments du modèle avec un couple  $l_k-l_l$  de segments de l'image est considéré comme localement cohérent s'il existe au moins un cercle, dans l'espace paramétrique défini par les écarts types (obtenus sur la diagonale de  $\Sigma_{rxy}$ ) autour de la moyenne  $\begin{bmatrix} \bar{r}_{kl} & \bar{x}_{kl} & \bar{y}_{kl} \end{bmatrix}^T$ , qui coupe le cercle d'incertitude sur la position.



**Figure 4.13** : propagation de l'incertitude des paramètres des primitives de l'image sur la zone de contrainte de la position avec la contrainte binaire.



**Figure 4.14** : prise en compte du bruit sur les paramètres du cercle de contrainte de la position

### Remarque

Rappelons que les zones d'erreur déterminées pour chaque contrainte ne sont que des approximations à l'ordre un des zones réelles. Nous considérons donc que le bruit sur les données image produit une variation faible sur les paramètres  $\rho$ ,  $d$  et  $v$  des segments observés ce qui permet de supposer la linéarité locale des équations utilisées.



## Evaluation de la méthode développée

Dans cette section sera évalué l'algorithme de mise en correspondance présenté. De même que pour les méthodes de calcul de la position et de l'orientation, l'algorithme sera testé avec des données synthétiques et sur des images réelles.

### Données synthétiques

#### Protocole

Suivant le même principe que les tests de la section précédente, des images synthétiques sont obtenues par projection des segments d'un modèle 3D à partir de positions et orientations générées de façon aléatoire dans les domaines de variation des paramètres de localisation. Un ensemble de 440 images ont donc été générées. Chaque image comprenait entre 5 et 12 segments correspondant aux segments 3D du modèle. Les données 2D ainsi obtenues ont été bruitées selon le même procédé que dans la section précédente.

Pour simuler les occultations et les défauts de détection du système de vision, une partie des segments projetés a été supprimée de la liste des segments de l'image. Pour simuler les segments correspondant aux objets non modélisés de chaque scène et aussi les détections dues au bruit du capteur (outliers), un ensemble de segments 2D générés également de façon aléatoire sont ajoutés à la liste des segments de la scène.

Sur chacune des images générées, 5 séries de tests ont été effectuées. La qualité d'initialisation pour chaque série est définie par les valeurs de  $Q_i$  regroupées dans la table 4.1. Le nombre de segments occultés varie de 1 à 3. Le nombre de segments ajoutés varie entre 10 et 15.

$Q_i$	1	2	3	4	5
$\delta r$ (m)	0.30	0.50	0.75	1.00	1.50
$\delta\phi$ (°)	10	15	20	30	40

*Table 4.1 : valeurs de  $Q_i$  utilisées dans les tests*

#### Résultats

Le premier résultat concerne le taux de réussite de l'algorithme à retrouver un nombre suffisant d'appariements corrects. Les résultats obtenus sont regroupés dans la table 4.2. Les chiffres correspondent à quatre catégories : succès, faux négatifs, faux positifs cohérents et faux positifs incohérents. On entend par succès les cas où l'hypothèse de mise en correspondance retenue comprend un nombre d'appariements corrects supérieur au seuil minimum considéré et aucun appariement incorrect. Les faux négatifs sont les cas où l'algorithme n'a pas réussi à réaliser un nombre suffisant d'appariements corrects. Les faux positifs correspondent aux cas où le nombre d'appariements corrects dépasse le nombre minimal requis mais où l'hypothèse de mise en correspondance contient des faux appariements. On distingue deux catégories de faux positifs, les faux positifs cohérents et les faux positifs incohérents. Les premiers sont des cas où les faux appariements sont dus à la nature ambiguë de la scène (segments très proches sur l'image). Dans ce cas l'erreur de localisation reste en dessous du seuil de précision toléré. Les faux positifs incohérents sont

des cas où l'hypothèse de mise en correspondance retenue comprend des faux appariements et où l'erreur de localisation est supérieure au seuil.

Le second résultats concernent les performances d'élagage de l'arbre d'interprétation par les contraintes unaire et binaire. Ceci se traduit par le nombre de combinaisons d'appariements retenues après la phase de génération d'hypothèses de mise en correspondance. C'est cette performance qui détermine l'efficacité de l'algorithme. La figure 4.15 représente le nombre d'hypothèses plausibles retenues après la phase de d'élagage, autrement dit le nombre d'hypothèses de mise en correspondance sélectionnées sans calcul de la position et de l'orientation. La figure représente aussi à titre indicatif le nombre maximal d'hypothèses de mise en correspondance. L'axe des abscisses représente les échantillons d'images testées. L'axe des ordonnées représente les nombres d'hypothèses retenues. Les graphes *a*, *b* et *c* montrent les résultats obtenus avec des qualités de l'estimation initiale de la position et de l'orientation correspondant respectivement aux valeurs 1,2 et 4 de  $Q_i$ . Les résultats montrent que pour une estimation initiale de 10° et de 30 cm ( $Q_i=1$ ), le nombre d'hypothèses de mise en correspondance reste inférieur à la centaine dans la quasi totalité des cas étudiés. Ce nombre évolue très faiblement pour une qualité d'initialisation moindre ( $Q_i=2$ ) il reste généralement inférieur à 100. Pour une erreur d'initialisation de 30° et 1 m ( $Q_i=3$ ), le nombre d'hypothèses ne dépasse les 400 que dans 11 cas sur les 440 étudiés.

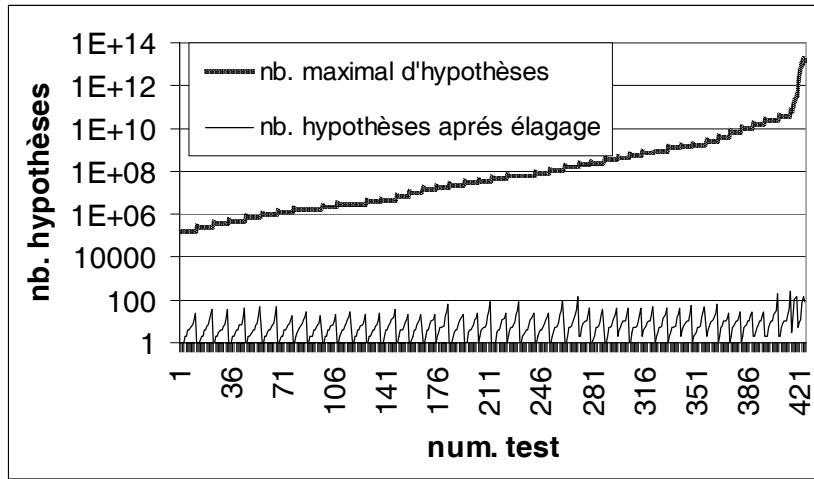
$Q_i$	1	2	3
Succès	421	418	415
Faux négatifs	5	5	5
Faux positifs cohérents	4	7	7
Faux positifs incohérents	0	0	3

*Table 4.2 : résultats de l'algorithme d'appariement*

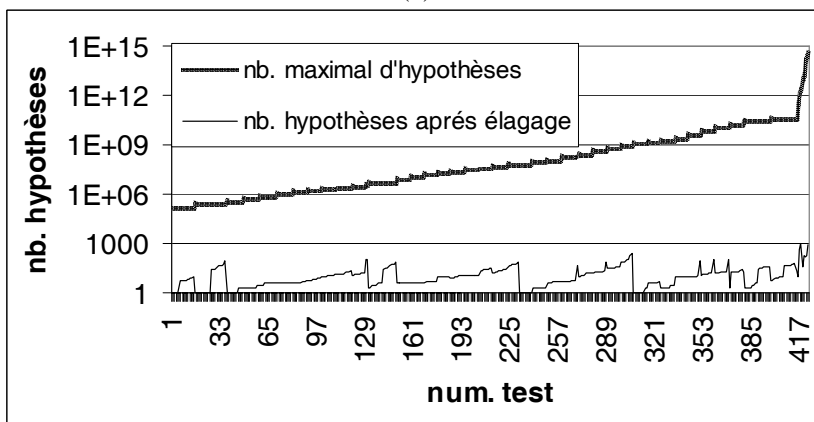
## Discussion

Les résultats montrent que pour une bonne estimation initiale de la position et de l'orientation, le nombre d'hypothèses de mise en correspondance reste très petit. Ce nombre évolue très faiblement lorsque la qualité d'initialisation diminue. Il reste de l'ordre de la centaine pour des marges d'erreur d'initialisation qui englobent largement les qualités d'estimation de la position et de l'orientation nécessaires dans le cadre d'une localisation incrémentale (de 30° et 1 m). Par ailleurs, ce nombre évolue faiblement en fonction du nombre maximal d'hypothèses qui, lui, croît exponentiellement en fonction du nombre de segments du modèle et de la scène. Ces deux propriétés sont déterminantes quant à l'efficacité de l'algorithme et son utilisation en ligne. Un autre constat que l'on peut faire est la persistance de cette qualité de sélection des hypothèses pour des initialisation très larges. Cette caractéristique peut être exploitée dans le cadre d'une localisation globale par exemple par répétition de la méthode sur différente zone de l'espace global. Une approche pour éviter les ambiguïtés doit cependant être associée à l'algorithme dans ce cas.

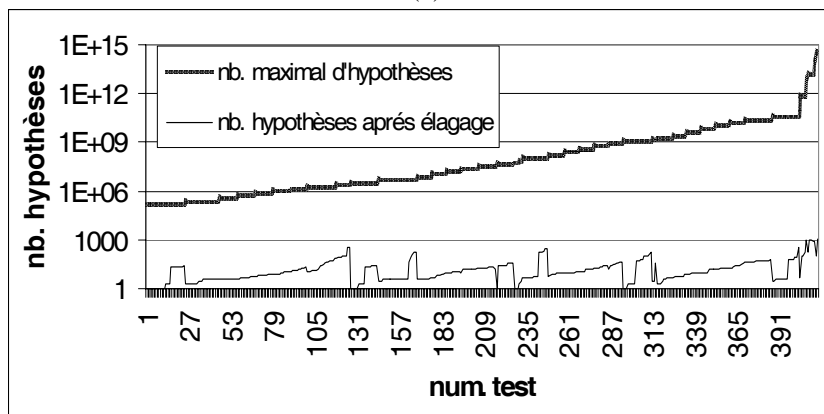
Un troisième constat concerne la capacité de sélection de la fonction de distance définie dans la section 4.2.2.3 et utilisée pour attribuer un score à chaque hypothèse de correspondance. Le taux très faible de faux positifs indique une bonne capacité de cette fonction à distinguer une cohérence réelle d'une hypothèse d'une cohérence apparente obtenue par des appariements accidentels.



(a)



(b)

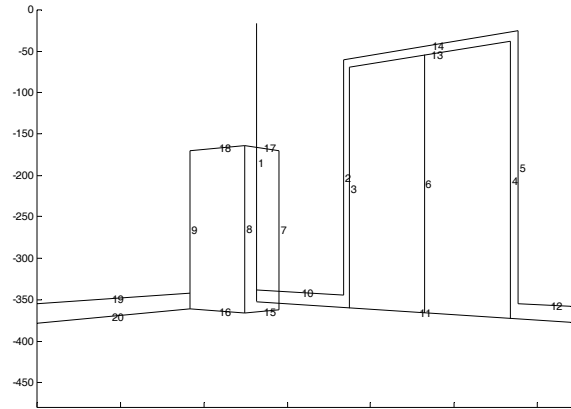


(c)

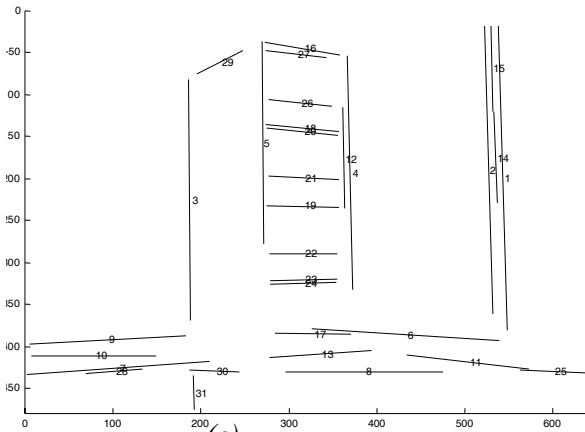
**Figure 4.15** : performances de l'élagage grâce aux contraintes unaire et binaire avec trois différents niveaux de qualité de l'estimation initiale de la position et de l'orientation.



(a)



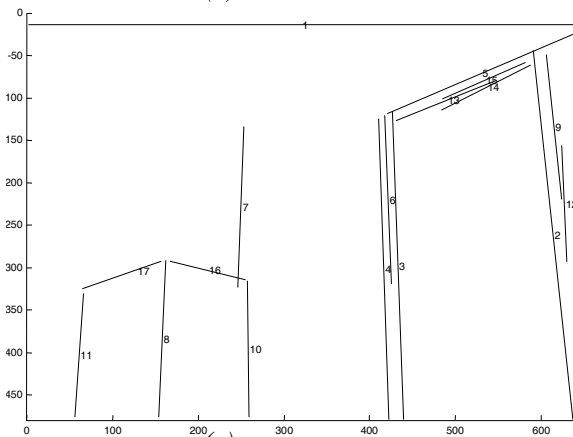
(b)



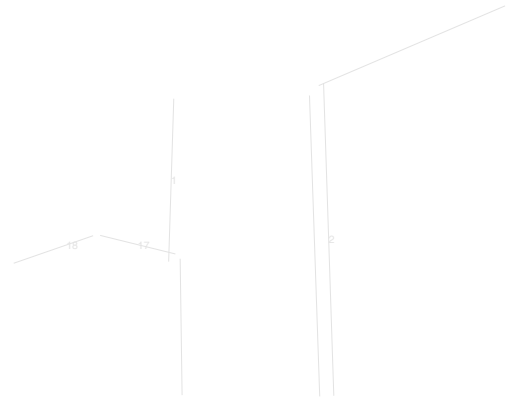
(c)



(d)



(e)



(f)

**Figure 4.17 :** (a) environnement utilisé pour les tests, (b) modèle correspondant, (c) et (e) exemples de scènes segmentées, (d) et (f) exemples de résultats d'appariements.

## Conclusions

Une méthode de mise en correspondance des segments de l'image 2D avec ceux du modèle 3D a été présentée. Cette méthode est basée sur une approche de recherche arborescente. Par rapport à l'approche classique composée d'une étape de génération d'hypothèses de correspondances et d'une étape de vérification et sélection de l'hypothèse la plus probable, certaines contributions ont été apportées.

Premièrement une étape de réduction de l'espace de recherche a été ajoutée. Elle consiste en la sélection d'un nombre réduit de segments candidats du modèle grâce aux régions d'invariance visuelle puis en la subdivision des ensembles de segments du modèle et de l'image chacun en deux sous ensembles. Deux arbres d'interprétation de dimensions inférieures sont alors construits.

Dans le cadre de la phase de génération d'hypothèse, deux contraintes géométriques d'ordre un et deux ont été définies pour tester la cohérence locale des appariements hypothétiques et élaguer les arbres d'interprétation. Ces contraintes sont directement applicables sur des appariements 2D-3D. La prise en compte de l'incertitude lors de l'application de ces contraintes a été également étudiée. Cette étape permet de générer un nombre réduit d'hypothèses de correspondance. La cohérence globale de ces hypothèses peut alors être testée par calcul de la transformation (rotation et translation de la caméra) correspondante et projection des segments du modèle.

Pour la phase de vérification, une fonction permettant de quantifier la ressemblance entre deux ensembles de segments 2D, inspirée de la distance de Hausdorff, a été définie. Elles sert à ordonner les hypothèses de correspondance retenues selon le degré de superposition des segments projetés du modèle et ceux observés sur l'image.

L'approche globale est flexible par rapport à la qualité de l'estimation initiale de la position permettant ainsi de moduler la fréquence d'échantillonnage du calcul en fonction de la disponibilité de données et des ressources logicielles.



## **Conclusion et perspectives**

### **Conclusions**

L'objectif de ce travail était d'apporter une contribution à la localisation de robot mobile dans un environnement intérieur partiellement modélisé grâce à la vision monoculaire. Pour cela, l'approche consistait à optimiser les performances de chaque étape composant le processus de localisation dans ce type d'approche. Au terme de notre travail nous pouvons tirer les conclusions suivantes :

#### **Sur les outils pour la localisation par vision monoculaire**

Cette partie concerne la modélisation et le calibrage de la caméra, la modélisation de l'environnement du robot et l'extraction de primitives de l'image. En se basant sur les états de l'art de chaque domaine, des techniques les plus appropriées à notre sens ont été retenues et adaptées au contexte. Concernant le calibrage, le choix a été motivé par le volume important de l'espace de travail. Une technique souple de calibrage a été choisie. Un traitement de correction des distorsions radiales est toutefois souhaitable dans cette échelle de grandeur.

Pour l'extraction des segments de l'image, une méthode simple et globale de détection de contours droits a été retenue. Le chaînage des contours est simplifié pour réduire le coût calculatoire.

L'environnement est modélisé par une représentation filaire. La gestion des occultations est gérée grâce à l'association d'une décomposition de l'espace navigable en régions d'invariance visuelle. Le modèle obtenu permet une expression simple et concise de l'environnement dans la mémoire du système. On suppose que l'environnement est délimité principalement par des surfaces verticales (hormis le sol). La gestion des occultations par des arêtes non verticales doit cependant être affinée.

#### **Sur le calcul de la position et de l'orientation**

Les principales motivations de notre travail dans ce domaine étaient :

- réduire les configurations singulières,
- assurer une bonne précision de localisation ,
- garantir la robustesse face à des conditions d'initialisation variables.

Pour cela les méthodes développées sont basées sur des techniques d'optimisation numériques et utilisent un modèle perspectif de caméra. Pour réduire les effets des non linéarités des équations, le nombre de degrés de liberté a été réduit par rapport aux formulations d'origines. Les fonctions d'erreur sont formulées directement en fonction des paramètres de position et d'orientation recherchés. Le nombre d'itérations est faible (4 à 5 itérations) ce qui rend les méthodes adaptées à une exécution en ligne. Les méthodes ayant été développées sous

Matlab, les résultats en terme de temps de calcul n'ont pu être présentés. La stratégie de calcul envisagée à la lumière des résultats obtenus consiste à appliquer la méthode basée sur l'approche de Phong-horaud à trois degrés de liberté puis à affiner le résultat avec quatre degrés de liberté. Les résultats obtenus montrent que la méthode permet pratiquement dans toutes les configurations d'obtenir une précision inférieure à  $3^\circ$  pour l'orientation et de l'ordre de 15 cm pour la translation.

### **Sur la mise en correspondance**

L'objectif fixé dans cette partie était de développer un algorithme de recherche de la bonne mise en correspondance entre les primitives de l'image et celles du modèle en prenant en compte certains aspects de notre contexte :

- on ne dispose le plus souvent que de vues partielles de chaque segment (à cause des occultations et des dimensions des segments),
- les effets de la perspective doivent être pris en compte,
- des segments non modélisés sont souvent présents dans la scène,
- l'algorithme doit être efficace pour une utilisation en ligne,
- la précision de l'estimation initiale est variable.

Pour remplir ce cahier des charges, la méthode développée est basée sur une approche arborescente. La phase préliminaire de réduction de l'espace de correspondance grâce aux régions d'invariance visuelle d'une part et à la subdivision de l'ensemble des segments de l'image d'autre part, permet d'optimiser les conditions pour la phase de recherche des deux arbres d'interprétation. Les contraintes de vérification de la cohérence locale des appariements sont directement applicables à des correspondances 2D-3D et prennent en compte la projection perspective. Elles sont d'ordre réduit (un et deux), ce qui permet d'éviter l'explosion exponentielle du nombre de cas à étudier. L'efficacité d'élagage est vérifiée par les résultats obtenus puisque le nombre d'hypothèses sélectionnées à partir de l'espace global des correspondances, et ce, sans calcul de la position et de l'orientation, est très faible. Les valeurs des temps d'exécution ne sont pas disponibles car les algorithmes sont développés sous Matlab. L'algorithme est robuste par rapport aux variations de la précision de l'estimation initiale (jusqu'à  $30^\circ$  et 1.5m). Cette propriété permet de réduire le taux d'échantillonnage et de reporter les phases de calcul pour des configurations où le risque d'erreur est élevé (pas assez de données ou saturation de données). Une fonction de vérification de la cohérence globale a également été définie. Les résultats montrent de bonnes performances de sélection de l'hypothèse la plus cohérente. Enfin, la gestion de l'incertitude a été abordée. Cependant, la précision des seuils (obtenus par une approximation des fonctions non linéaires de contrainte de la position par les moments d'ordre un et deux) peut être améliorée.

## Perspectives

Dans les conclusions de ce mémoire, plusieurs points ont été soulignés comme devant faire l'objet d'études plus approfondies.

D'abord, d'un point de vue pratique, l'implantation des algorithmes développés sur un robot réel et avec un langage temps réel permettrait d'évaluer de façon plus précise les performances en termes de temps d'exécution.

Ensuite, concernant l'aspect théorique du travail, la détermination des seuils d'incertitude dans les contraintes unaires et binaires (section 4.2.3) a été effectuée grâce à la propagation des moments du premier et second ordre des équations de contraintes. Ces fonctions étant fortement non-linéaires, une estimation par un développement en termes d'ordre plus élevé donnerait une estimation plus « serrée » des seuils sur les paramètres de contrainte. Ceci aurait un effet bénéfique sur les performances d'élagage des contraintes unaire et binaire. En effet, une sélection plus importante des hypothèses d'appariements pourrait être effectuée sans risquer de « perdre » des appariements corrects.

Un autre module reste également à développer. Il s'agit d'une procédure de localisation globale. En effet, si une défaillance du système odométrique survient ou lors de la mise en service du robot, il serait difficile d'appliquer notre méthode de mise en correspondance. Toutefois, la faible sensibilité de cet algorithme à la variation de la qualité d'initialisation permettent d'envisager son utilisation pour une localisation globale. Une solution serait le « multi-tracking » qui consiste à répéter l'algorithme sur différentes zones de l'espace global. La contrainte temporelle doit dans ce cas être assouplie. Les régions d'invariance visuelle peuvent servir alors de base pour le clustering de l'espace des positions et des orientations. Cette approche peut toutefois souffrir des ambiguïtés qui peuvent survenir à cause des symétries du modèle.

Un autre solution serait d'associer, dans ce cas, notre algorithme à une méthode d'estimation grossière de la position et de l'orientation du robot. Des méthodes de reconnaissance de vues panoramiques peuvent servir à cet effet ([Mouaddib00]). L'utilisation d'une base d'images indexée peut également être envisagée. Le robot saisit des vues lors d'une phase d'apprentissage et leur associe une position et une orientation. Lorsque le robot est complètement perdu dans son environnement, il peut lancer une procédure de recherche en multipliant les prises de vues jusqu'à identifier dans la base une image qui a une signature suffisamment proche de la vue courante ([Lee94]). Dans [Gaussier00], une approche n'utilisant pas les connaissances du modèle est présentée. Elle permet au robot de rejoindre un endroit déjà exploré par une méthode neuronale proche du système de navigation du règne animal. Des points de l'environnement où la méthode de localisation a le plus de chance de réussite peuvent alors être définis comme objectifs. En cas de défaillance de la localisation incrémentale, le robot peut alors lancer une procédure pour rejoindre un de ces objectifs.

# Annexe A

## Calcul des matrices jacobiennes pour la contrainte unaire

On a deux fonctions  $f_\phi$  et  $f_{ABC}$  telles que :

$$\begin{cases} \phi = f_\phi([\rho, d]^T) \\ [A, B, C]^T = f_{ABC}([\rho, d]^T) \end{cases}$$

$f_\phi$  est définie par :

$$\phi = \arctan\left(\frac{b}{a}\right) \pm \arccos\left(\frac{c}{\sqrt{a^2 + b^2}}\right)$$

avec :

$$\begin{cases} a = a_1 n_x + a_2 n_y + a_3 n_z \\ b = b_1 n_x + b_2 n_y + b_3 n_z \\ c = c_1 n_x + c_2 n_y + c_3 n_z \end{cases}$$

et

$$\begin{cases} a_1 = v_x \cos(\theta) \\ a_2 = v_y \\ a_3 = -v_z \sin(\theta) \end{cases}, \quad \begin{cases} b_1 = -v_y \cos(\theta) \\ b_2 = v_x \\ b_3 = v_z \cos(\theta) \end{cases} \quad \text{et} \quad \begin{cases} c_1 = v_z \cos(\theta) \\ c_2 = 0 \\ c_3 = v_z \cos(\theta) \end{cases}$$

L'équation d'une droite de l'image s'écrit  $u \cos(\rho) + v \sin(\rho) - d = 0$

En remplaçant  $u$  et  $v$  par leurs expressions en fonction des coordonnées caméra on obtient

$$\left(-\alpha \frac{x}{z} + u_0\right) \cos(\rho) + \left(\alpha \frac{x}{z} + v_0\right) \sin(\rho) - d = 0$$

expression que l'on réécrit

$$x \cos(-\rho) + y \sin(-\rho) + z \frac{1}{\alpha} (u_0 \cos(\rho) + v_0 \sin(\rho) - d) = 0$$

Ceci n'est autre que l'équation du plan d'interprétation du segment observé. Il en découle que

$$\begin{cases} n_x = \cos(\rho) \\ n_y = -\sin(\rho) \\ n_z = \frac{1}{\alpha} (d - u_0 \cos(\rho) - v_0 \sin(\rho)) \end{cases}$$

Les dérivées partielles sont :

$$\left\{ \begin{array}{l} \frac{\partial \phi}{\partial \rho} = \left( \frac{1}{1 + \left(\frac{b}{a}\right)^2} \cdot \frac{a \frac{\partial b}{\partial \rho} - b \frac{\partial a}{\partial \rho}}{a^2} \right) \pm \left( \frac{a^2 + b^2}{a^2 + b^2 - c^2} \cdot \frac{\frac{\partial c}{\partial \rho} - c \left( a \frac{\partial a}{\partial \rho} + b \frac{\partial b}{\partial \rho} \right)}{\sqrt{a^2 + b^2}} \right) \\ \frac{\partial \phi}{\partial d} = \left( \frac{1}{1 + \left(\frac{b}{a}\right)^2} \cdot \frac{a \frac{\partial b}{\partial d} - b \frac{\partial a}{\partial d}}{a^2} \right) \pm \left( \frac{a^2 + b^2}{a^2 + b^2 - c^2} \cdot \frac{\frac{\partial c}{\partial d} - c \left( a \frac{\partial a}{\partial d} + b \frac{\partial b}{\partial d} \right)}{\sqrt{a^2 + b^2}} \right) \end{array} \right.$$

avec

$$\left\{ \begin{array}{l} \frac{\partial a}{\partial \rho} = a_1 \frac{\partial n_x}{\partial \rho} + a_2 \frac{\partial n_y}{\partial \rho} + a_3 \frac{\partial n_z}{\partial \rho} \\ \frac{\partial b}{\partial \rho} = b_1 \frac{\partial n_x}{\partial \rho} + b_2 \frac{\partial n_y}{\partial \rho} + b_3 \frac{\partial n_z}{\partial \rho} \\ \frac{\partial c}{\partial \rho} = c_1 \frac{\partial n_x}{\partial \rho} + c_2 \frac{\partial n_y}{\partial \rho} + c_3 \frac{\partial n_z}{\partial \rho} \end{array} \right. \text{ et } \left\{ \begin{array}{l} \frac{\partial a}{\partial d} = a_1 \frac{\partial n_x}{\partial d} + a_2 \frac{\partial n_y}{\partial d} + a_3 \frac{\partial n_z}{\partial d} \\ \frac{\partial b}{\partial d} = b_1 \frac{\partial n_x}{\partial d} + b_2 \frac{\partial n_y}{\partial d} + b_3 \frac{\partial n_z}{\partial d} \\ \frac{\partial c}{\partial d} = c_1 \frac{\partial n_x}{\partial d} + c_2 \frac{\partial n_y}{\partial d} + c_3 \frac{\partial n_z}{\partial d} \end{array} \right.$$

et avec :

$$\left\{ \begin{array}{l} \frac{\partial n_x}{\partial \rho} = -\sin(\rho) \\ \frac{\partial n_y}{\partial \rho} = -\cos(\rho) \\ \frac{\partial n_z}{\partial \rho} = \frac{1}{\alpha} (u_0 \sin(\rho) - v_0 \cos(\rho)) \end{array} \right. \text{ et } \left\{ \begin{array}{l} \frac{\partial n_x}{\partial d} = 0 \\ \frac{\partial n_y}{\partial d} = 0 \\ \frac{\partial n_z}{\partial d} = \frac{1}{\alpha} \end{array} \right.$$

$f_{ABC}$  est définie comme suit

$$\left\{ \begin{array}{l} A = A_1 n_x \cos(\phi) + A_2 n_y \sin(\phi) + A_3 n_z \cos(\phi) \\ B = B_1 n_x \sin(\phi) + B_2 n_y \cos(\phi) + B_3 n_z \sin(\phi) \\ C = [C_1 n_x + C_2 n_y + C_3 n_z] \cos(\phi) + [C_4 n_x + C_5 n_y + C_6 n_z] \sin(\phi) + [C_7 n_x + C_8 n_y + C_9 n_z] \end{array} \right.$$

avec comme valeurs des constantes :

$$\left\{ \begin{array}{l} A_1 = -\cos(\theta) \\ A_2 = -1 \\ A_3 = \sin(\theta) \end{array} \right. \left\{ \begin{array}{l} B_1 = \cos(\theta) \\ B_2 = -1 \\ B_3 = -\sin(\theta) \end{array} \right. \left\{ \begin{array}{l} C_1 = x \cos(\theta) \\ C_2 = y \\ C_3 = x \sin(\theta) \\ C_4 = -y \cos(\theta) \\ C_5 = x \\ C_6 = y \sin(\theta) \\ C_7 = (z - Z) \sin(\theta) \\ C_8 = 0 \\ C_9 = (z - Z) \cos(\theta) \end{array} \right.$$

où  $(x,y,z)$  sont les coordonnées d'un point quelconque du segment du modèle et  $Z$  la hauteur du centre optique.

Nous obtenons :

$$\left\{ \begin{array}{l} \frac{\partial A}{\partial \rho} = A_1 \left( \frac{\partial n_x}{\partial \rho} \cos(\phi) + n_x \frac{\partial \cos(\phi)}{\partial \rho} \right) + A_2 \left( \frac{\partial n_y}{\partial \rho} \sin(\phi) + n_y \frac{\partial \sin(\phi)}{\partial \rho} \right) \frac{\partial}{\partial \rho} \\ \hspace{18em} + A_3 \left( \frac{\partial n_z}{\partial \rho} \cos(\phi) + n_z \frac{\partial \cos(\phi)}{\partial \rho} \right) \\ \frac{\partial B}{\partial \rho} = B_1 \left( \frac{\partial n_y}{\partial \rho} \sin(\phi) + n_y \frac{\partial \sin(\phi)}{\partial \rho} \right) + B_2 \left( \frac{\partial n_x}{\partial \rho} \cos(\phi) + n_x \frac{\partial \cos(\phi)}{\partial \rho} \right) \\ \hspace{18em} + B_3 \left( \frac{\partial n_z}{\partial \rho} \cos(\phi) + n_z \frac{\partial \cos(\phi)}{\partial \rho} \right) \\ \frac{\partial C}{\partial \rho} = \left( C_1 \frac{\partial n_x}{\partial \rho} + C_2 \frac{\partial n_y}{\partial \rho} + C_3 \frac{\partial n_z}{\partial \rho} \right) \cos(\phi) + (C_1 n_x + C_2 n_y + C_3 n_z) \frac{\partial \cos(\phi)}{\partial \rho} \\ \hspace{8em} + \left( C_4 \frac{\partial n_x}{\partial \rho} + C_5 \frac{\partial n_y}{\partial \rho} + C_6 \frac{\partial n_z}{\partial \rho} \right) \cos(\phi) + (C_4 n_x + C_5 n_y + C_6 n_z) \frac{\partial \sin(\phi)}{\partial \rho} \\ \hspace{8em} + \left( C_7 \frac{\partial n_x}{\partial \rho} + C_8 \frac{\partial n_y}{\partial \rho} + C_9 \frac{\partial n_z}{\partial \rho} \right) \end{array} \right.$$

Les dérivées par rapport à  $d$  s'obtiennent par un calcul identique.

## Calcul des matrices jacobiennes pour la contrainte binaire

- Rayon du cercle de contrainte :

$$r_{kl} = \left\| \overrightarrow{p_i p_j} \right\| \frac{1}{2 \sin(\omega_{kl})}$$

avec

$$\omega_k = \arctan\left(\frac{v_k - v_0}{\alpha_v}\right) - \arctan\left(\frac{v_l - v_0}{\alpha_v}\right)$$

$$\begin{cases} (x - x_j)^2 + (y - y_j)^2 - 4r_{kl}^2 = 0 \\ \cos(\rho_{ij})x + \sin(\rho_{ij})y - d_{ij} - r_{kl} \cos(\omega_{kl}) = 0 \end{cases}$$

$$x = ar_{kl} \cos(\omega_{kl}) + b \pm \sqrt{c(r_{kl} \cos(\omega_{kl}))^2 + dr_{kl} \cos(\omega_{kl}) + e}$$

où  $a, b, c, d$  et  $e$  sont des constantes.

- Coordonnées de  $x_c$  et  $y_c$  du centre du cercle de contrainte

$$\begin{cases} x_c = \frac{x + x_i}{2} \\ y_c = \frac{y + y_i}{2} \end{cases}$$

avec

$$x = ar_{kl} \cos(\omega_{kl}) + b \pm \sqrt{c(r_{kl} \cos(\omega_{kl}))^2 + dr_{kl} \cos(\omega_{kl}) + e}$$

Matrice Jacobienne  $\mathcal{J}_{f_{xy}}$  recherchée :

$$\mathcal{J}_{f_{xy}} = \begin{bmatrix} \frac{\partial r_{kl}}{\partial v_k} & \frac{\partial r_{kl}}{\partial v_l} \\ \frac{\partial x_{kl}}{\partial v_k} & \frac{\partial x_{kl}}{\partial v_l} \\ \frac{\partial y_{kl}}{\partial v_k} & \frac{\partial y_{kl}}{\partial v_l} \end{bmatrix}$$

Les points principaux nécessaires au calcul sont :

$$\frac{\partial r_{kl}}{\partial v_k} = \left\| \overrightarrow{p_i p_j} \right\| \frac{\cos(\omega_{kl})}{2(\sin(\omega_{kl}))^2} \frac{\partial \omega_{kl}}{\partial v_k}$$

$$\frac{\partial \omega_{kl}}{\partial v_k} = \frac{\left\| \overrightarrow{p_i p_j} \right\|}{\alpha_v} \frac{\cos(\omega_{kl})}{1 + \left(\frac{v_k - v_0}{\alpha_v}\right)^2}$$

Le reste des dérivées sont des compositions de ces deux termes.

## Annexe B

### Coordonnée de l'intersection d'un segment image avec la ligne d'horizon pour $\theta = 90^\circ$ à partir d'une image avec $\theta \neq 90^\circ$

Soit  $l_i$  un segment de l'image satisfaisant la contrainte d'orientation définie en section 4.2.2.2.3 (autrement dit pouvant potentiellement correspondre à un segment vertical dans le repère du modèle) observé par la caméra alors que l'orientation est donnée par la matrice  $R(\phi, \theta, 0)$  et la translation par le vecteur  $T([x \ y \ z]^t)$ . Soit  $v_{hi}$  l'abscisse du point d'intersection de  $l_i$  avec la ligne d'horizon définie en section 4.2.2.2. Ce point de l'image est en fait la projection d'un point 3d  $p_i(x_i, y_i, z_i)$  appartenant au segment correspondant du modèle.  $p_i$  se projette sur l'image comme suit

$$(x_i', y_i', z_i') = R(p_i - T)$$

puis

$$(u_{hi}, v_{hi}) = (\alpha_u \cdot \frac{x_i'}{z_i'} + u_0, \alpha_v \cdot \frac{y_i'}{z_i'} + v_0)$$

Sachant que  $z_i = z$  on obtient

$$v_{hi} = \frac{\alpha_v \cdot ((x_i - x) \cdot \sin(\phi) + (y_i - y) \cdot \cos(\phi))}{(-(x_i - x) \cdot \cos(\phi) + (y_i - y) \cdot \sin(\phi)) \cdot \sin(\theta)} + v_0$$

Le but est de calculer la mesure  $v_i$  correspondant  $v_{hi}$  en supposant que  $\theta$  est égal à  $90^\circ$  :

$$v_i = \frac{\alpha_v \cdot ((x_i - x) \cdot \sin(\phi) + (y_i - y) \cdot \cos(\phi))}{-(x_i - x) \cdot \cos(\phi) + (y_i - y) \cdot \sin(\phi)} + v_0$$

Il s'en suit

$$v_i = (v_{hi} - v_0) \cdot \sin(\theta) + v_0$$

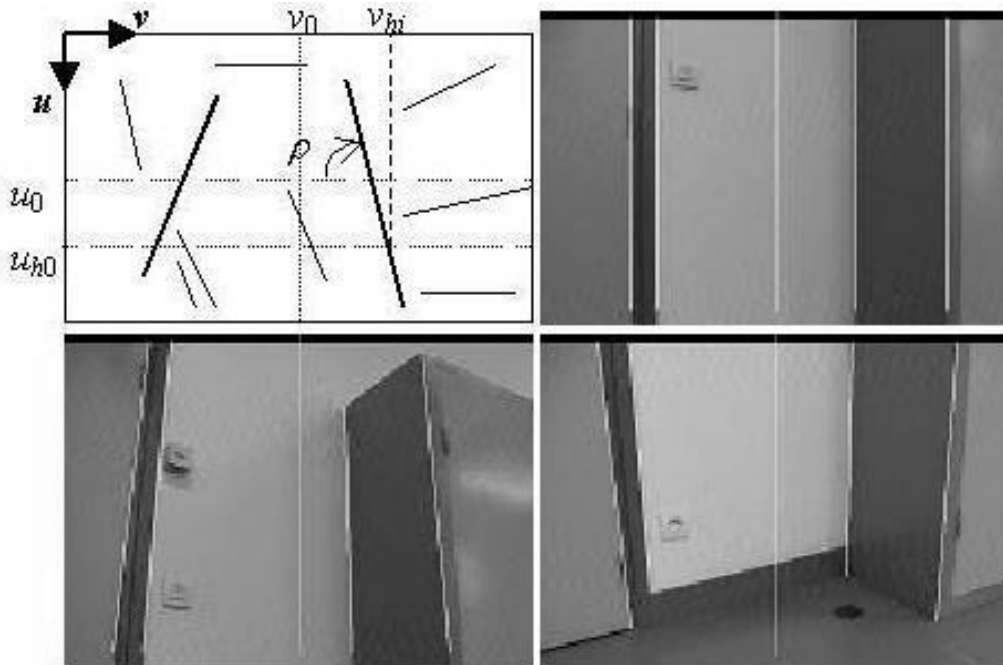


## Détection des segments image pouvant correspondre à des segments verticaux du modèle

Soit  $l$  un segment de l'image défini par sa pente  $\rho$  et sa distance à l'origine du repère  $d$ .  $l$  peut être mis en correspondance avec un segment vertical du modèle si les règles suivantes sont vérifiées :

- si  $0 < \theta < 90^\circ$  alors
  - $\rho_{min} < \rho < \rho_{max}$  lorsque  $l$  est contenu dans la partie gauche de la ligne  $v=v_0$
  - ou  $-\rho_{max} < \rho < -\rho_{min}$  lorsque  $l$  est contenu dans la partie droite de la ligne  $v=v_0$ .
- si  $90^\circ < \theta < 180^\circ$  alors
  - $-\rho_{max} < \rho < -\rho_{min}$  lorsque  $l$  est contenu dans la partie droite de la ligne  $v=v_0$
  - ou  $\rho_{min} < \rho < \rho_{max}$  lorsque  $l$  est contenu dans la partie gauche de la ligne  $v=v_0$ .

$\rho_{min}$  et  $\rho_{max}$  sont proportionnels à  $\theta_{min}$  et  $\theta_{max}$  qui est généralement compris entre  $45^\circ$  et  $135^\circ$  respectivement (figure ci-dessous).



## Annexe C

### Ordre d'apparition de droite à gauche des segments verticaux sur l'image

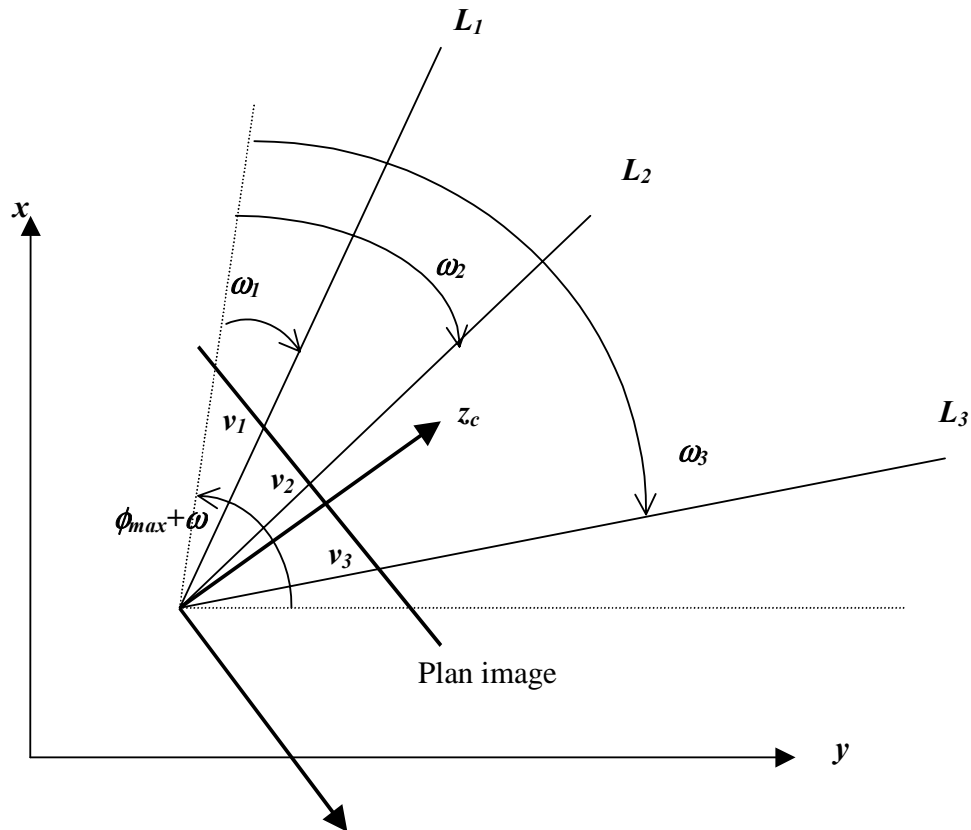
Soit un ensemble de segments verticaux  $\{L_1, L_2, \dots, L_i\}$  du modèle et leurs intersections respectives avec la ligne d'horizon (chapitre 4)  $\{v_1, v_2, \dots, v_i\}$ . Soit  $[\phi_{\min}, \phi_{\max}]$  une estimation de l'orientation de la caméra et  $\omega$  son angle d'ouverture (figure ci-dessous).

On calcule l'angle de vue de chaque segment  $L_i$  comme suit :

$$\omega_i = \arctan\left(\frac{v_i - v_0}{\alpha_v}\right) - \phi_{\max} - \omega$$

L'ordre d'apparition de droite à gauche des segments est suivant l'ordre décroissant des valeurs  $\omega_i$ .

Lorsque la caméra se déplace dans une même région d'invariance visuelle, les valeurs de  $\omega_i$  changent mais l'ordre reste le même.



# Références

## Références personnelles

[Ait Aider02] O. Ait Aider, P. Hoppenot, E. Colle, « A Model to Image Straight Line Matching Method for Vision-Based Indoor Mobile Robot Self-Location », In Proc. of the 2002 IEEE/RSJ Int. Conference on Intelligent Robots and Systems IROS'2002, Lausanne, pp. 460-465, 2002.

[Ait Aider02] O. Ait-Aider, P. Hoppenot, E. Colle, « Adaptation of Lowe's camera pose recovery algorithm to mobile robot self-localisation », Robotica 2002, Vol. 20, pp. 385-393, 2002.

[Ait Aider01] O. Ait Aider, P. Hoppenot, E. Colle, « Localisation by camera of a rehabilitation robot », ICORR 7th Int. Conf. On Rehab. Robotics, France, pp. 168-176, 25-27 avril 2001.

[Ait Aider00] O. Ait Aider, P. Hoppenot, E. Colle, « Localisation caméra/objet par correspondances point 3D / pixel - application à la localisation d'un robot mobile », JJCR 13, Rennes, 28 et 29 Septembre 2000.

[Rybarczyk02] Y. Rybarczyk, O. Ait Aider, P. Hoppenot, E. Colle, « Commande d'un système d'assistance robotique aux personnes handicapées », Handicap'2002, pp. 81-87, Paris, 13-14 juin 2002.

[Hoppenot01] P. Hoppenot, E. Colle, O. Ait Aider, Y. Rybarczyk, « ARPH - Assistant Robot for Handicapped People - A pluridisciplinary project » IEEE Roman'2001, Bordeaux and Paris, pp. 624-629, 18-21 Sept 2001.

## Autres références

[Abidi95] M. A. Abidi, T. Chandra, « A New Efficient and Direct Solution for Pose Estimation using Quadrangular targets : Algorithm and Evaluation », IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 17, N°5, 1995, pp. 534-538.

[Araujo96] H. Araujo, R. Carceroni, C. Brown, « A fully projective formulation for Lowe's tracking algorithm », Technical report 641, Univ. of Rochester, 1996.

[Ayache86] N. Ayache and O. Faugeras and O.D. Hyper, « A New Approach for the Recognition and Positioning of Two-Dimensional Objects », IEEE Trans. on Pattern Analysis and Machine Intelligence, 8(1), 1986, pp. 44-54.

[Barshan95] B. Barshan and H.F. Durrant-White, « Inertial navigation system for a mobile robot », IEEE Trans on Robotics and Automation, Vol 12, N° 5, pp. 328-342, 1995.

[Batista99] J. Batista, H. Araujo and A.T. de Almeida, « Iterative multistep camera calibration », IEEE Trans. on Robotics and Automation, Vol. 15, N°5, pp. 897-917, 1999.

[Besl92] P.J. Besl and N.A. MacKay, « A method for registration of 3-D shapes », IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 14, N°2, pp. 239-256, 1992.

[Betke98] M. betke and L. Gurvits, « Mobile robot localization using landmarks », IEEE Trans. on Robotics and Automation, Vol. 13, N°2, pp. 251-263, 1998.

[Bolles82] R.C. Bolles and R.A. Cain, « Recognizing and locating partially visible objects : the locus point-feature-focus method », Int. Journal Robotics Research, Vol. 1, pp57-82, 1982.

[Borenstein91] J. Borenstein and Y. Koren, « Histogram in-motion mapping for mobile robot obstacle avoidance », IEEE Trans. on Robotics and Automation, Vol. 7, N°4, pp. 535-539, 1991.

- [Borenstein97] J. Borenstein, H.R. Everett, L. Feng and D. Wehe, « Mobile Robot Positioning: Sensors and Techniques », *Journal of Robotic Systems*, 14(4), 1997, pp. 231-249.
- [Borges00] G. A. Borges and M. J. Aldon, « Motion estimation by iterative 2-D features matching in range images », In *proc. IEEE international Conference on Robotics and Automation*, 2000.
- [Borges02] G. A. Borges and M. J. Aldon, « Optimal mobile robot pose estimation using geometrical maps », *IEEE Trans. On Robotics and Automation*, Vol 18, N°1, pp. 87-94, 2002.
- [Bowyer92] K. W. Bowyer, « Workshop panel report – why aspect graphs are not (yet) paractical for computer vision », *IEEE/SMC'93, Systems, Man and Cybernetics*, pp. 294-299, Le Touquet, France, 1993.
- [Breuel92] T.M. Breuel, « Geometric aspects of visual object recognition. Doctoral dissertation, AI-TR 1374, MIT, 1992.
- [Breuel92] T.M. Breuel, « Fast recognition using adaptative subdivisions of transformation space », In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 445-451, 1992.
- [Byrne92] R.H. Byrne, P.R. Klarer and J.B. Pletta, « Techniques for autonomous navigation », *Sandia Report SAND92-0457*, Sandia National Laboratories, Albuquerque, NM, 1992.
- [Carceroni97] R. L. Carceroni, C. M. Brown, « Numerical Methods for Model-Based Pose Recovery », *Technical Report 659*, Université de Rochester, Computer Science Department, New-York, 1997.
- [Carceroni97] R.L. Carceroni and C.M. Brown, « Numerical methods for model-based pose recovery », *Technical report 659*, University of Rochester, Computer Science Department, New York, 1997.
- [Chang99] C.C. Chang and W.H. Tsai, « Reliable Determination of Object Pose from Line Features by Hypothesis Testing », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(12), 1999, pp. 1235-1241.
- [Chatila85] R. Chatila and J.P. Laumond, « Position referencing and consistent world modeling for mobile robots », *Proc of the IEEE International Conference on Robotics and Automation (ICRA85)*, pp. 138-145, 1985.
- [Christy98] S. Christy, R. Horaud, « Iterative Pose Computation from Line Correspondences », *Technical Report, INRIA98*.
- [Cox91] J. Cox : Blanche, « An experiment in guidance and navigation of autonomous robot vehicle », *IEEE Trans on Robotics and Automation*, Vol 7, N°2, Avril 1991.
- [Dementhon95] D. F. Dementhon, L. S. Davis : « Model-based object pose in 25 lines of code correspondences », *Int. J. of Computer Vision*, Vol. 15, 1995, pp. 123-141.
- [DeSouza02] G. N. DeSouza and A. C. Kak, « Vison for mobile robot navigation : a survey », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol 24, N°2, 2002, pp. 237-267.
- [Dhome89] M. Dhome, M. Richetin, J.T. Lapresté, G. Rives : « Determination of the attitude of 3-D objects from single perspective view », *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 11, N°12, 1989, pp. 1256-1278.
- [Dombre88] E. Dombre, W. Khalil, « Modélisation et commande des robots », *Hermes*, 1988.
- [Du96] F. Du and M. Brady, « Self-calibration of the intrinsic parameters of cameras for active vision », *Proc. Of the IEEE Conf. Comput. Vision Pattern Recognition*, pp. 477-482, 1993.
- [Faugeras86] O.D. Faugeras, G. Toscani, « The calibration problem for stereo », *Proc. Second Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.321-334, 1986.
- [Faugeras93] O.D. Faugeras : « three dimensional computer vision : a geometric viewpoint », *MIT Press, Boston*, 1993.

- [Fischer92] D. Fischer, O. Bachar, R. Nussinov and H. Wolfson, « An efficient automated computer vision-based technique for detection of three dimensional structural motifs in proteins », *J. of Biomolecular structures and dynamics*, Vol. 9, N°4, pp. 769-789, 1992.
- [Fischler81] M. A. Fischler, R. C. Bolles : « Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography », *Communication of the ACM*, Vol. 24, N°6, pp. 381-395, 1981.
- [Grimson87] W.E.L. Grimson and T. Lozano-Perez, « Localizing Overlapping Parts by Searching the Interpretation Tree », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4), pp. 469-481, 1987.
- [Grimson90] W.E.L. Grimson and D.P. Huttenlocher, « On the Sensitivity of the Hough Transform for Object Recognition », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(3), pp. 255-274, 1990.
- [Grimson90] W.E.L. Grimson and D.P. Huttenlocher, « On the sensitivity of geometric hashing », *In Proc. Third ICCV*, pp. 334-338, 1990.
- [Grimson90b] W.E.L. Grimson, « Object Recognition: The Role of geometric constraints », MIT Press, 1990.
- [Grimson91] W.E.L. Grimson and D.P. Huttenlocher, « On the Verification of Hypothesized Matches in Model-Based Recognition », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(12), pp. 1202-1213, 1991.
- [Grimson94] W.E.L. Grimson and D.P. Huttenlocher and D. W. Jacobs, « A study of affine matching with bounded sensor error », *Int. Journal of computer vision*, Vol. 13, N° 1, pp. 7-32, 1994.
- [Guibas92] L. Guibas, R. Motwani and P. Raghavan, « The robot localization problem in two dimensions », *In Proc. Third ACM-SIAM Symposium on Discrete Algorithms*, pp. 259-268, 1992.
- [Hager90] G. Hager and M. Mintz, « Sensor modeling and robust sensor data fusion », *proc. Of the International Symposium on Robotics Research*, pp.69-74, 1990.
- [Hartley94] R. Hartley, « Self-Calibration from multiple views with a rotating camera », *Proc. Third European Conf. Computer Vision*, 1994.
- [Hoppenot97] P. Hoppenot, « Contribution de la robotique mobile à l'assistance aux personnes handicapées », Thèse de doctorat, Université d'Evry Val d'Essonne (EVE), 27 nov. 1997.
- [Hoppenot98] P. Hoppenot, E. Colle, « Real-time localisation of a low-cost mobile robot with poor ultrasonic data », *IFAC journal, Control Engineering practice*, vol. 6, pp.925-934, 1998.
- [Hoppenot01] P. Hoppenot, E. Colle, « Localisation and control of a rehabilitation robot by close human-machine co-operation », *IEEE Transaction on Neural System and Rehabilitation Engineering*, vol. 9, pp. 181-190, June 2001.
- [Horaud89] R. Horaud, B. Conio, O. Leboulleux, B. Lacolle, «A Analytic Solution for the Perspective 4-Point Problem », *Computer Vision, Graphics and Processing*, 47, 33-44, 1989.
- [Horaud93] R. Horaud, O. Monga, « Vision par ordinateur, outils fondamentaux », Hermes, Paris, France, 1993.
- [Horaud94] R. Horaud, S. Christy, F. Dornaika, « Object pose : the link between weak perspective, para perspective and full perspective », *Technical report RR-2356, INRIA*, 1994.
- [Ikeuchi88] K. Ikeuchi and T. Kanade, « Automatic generation of object recognition programs », *Proceedings IEEE*, Vol. 76, N°8, pp. 1016-1035, 1988.
- [Ishii89] M. Ishii, S. Sakane, M. Kakikura, Y. Mikami, « A 3-D sensor system for teaching robot paths and environments », *International Journal of Robotics Researchs*, 6(2), 45-59, 1989.

- [Jensfelt01] P. Jensfelt and S. Kristensen, « Active global localization for a mobile robot using multiple hypothesis tracking », *IEEE Trans. on Robotics and Automation*, Vol. ,N° , pp. - , 2001.
- [Kamgar-Parsi97] B. Kamgar-Parsi and B. Kamgar-Parsi, « Matching sets of 3D line segments with application to polygonal arc matching », *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 19, N°10, pp. 1090-1099, 1997.
- [Kosaka92] A. Kosaka and A.C. Kak, « fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties », *Computer Vision, Graphics and Image Processing – Image understanding*, Vol. 56, N°3, pp.271-329, 1992.
- [Kuc87] R. Kuc and M. Siegel, « Physically based simulation model for acoustic sensor robot navigation », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol 9, N°6, 1987, pp. 766-778
- [Lamdan88] Y. Lamdan and H.J. Wolfson, « Geometric hashing: A General and Efficient Model-Based Recognition Scheme », *Proc. of Second ICCV*, 1988, pp. 238-289.
- [Lamdan91] Y. Lamdan and H.J. Wolfson, « On the error analysis of geometric hashing », In *IEEE Int. Conf. On Comput. Vision and Pattern Recognition*, pp. 22-27, 1991.
- [Lee94] P. S. Lee, Y. E. Shen, L. L. Wang « Model-based location of automated guided vehicles in the navigation session by 3D computer vision », *Journal of robotic systems*, Vol. 11, N°3, 1994, pp. 181-195.
- [Leonard91] J. Leonard and H. Durrant-White, « Mobile robot localization by tracking geometric beacons », *IEEE Trans on Robotics and Automation*, Vol 7, N° 3, pp. 376-382, 1991.
- [Linnainmaa88] S. Linnainmaa, D. Harwood, L. S. Davis, « Pose determination of a Three-dimensional Object using Triangle Pairs », *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 10, N°5, pp. 634-647, 1988.
- [Liu90] Y. Liu, T.S. Huang, O.D. Faugeras, « Determination of camera location from 2D to 3D line and point correspondences », *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 12, N°1, pp. 28-37, 1990.
- [Lowe85] D. G. Lowe, « Perceptual organization and visual recognition », Boston, MA: Kluwer, 1985, ch 7.
- [Lowe87] D. G. Lowe , « Three-dimensional object recognition from single two dimensional images », *Artificial Intelligence*, vol. 31, N°3, pp. 355-395, 1987.
- [Lowe91] D. G. Lowe, « Fitting parametrized three-dimensional models to images », *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 13, N°5, pp. 441-450, 1991.
- [Lu00] C. P. Lu, G. D. Hager, E. Mjølness, « Fast and globally convergent pose estimation from video images », *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 22, N°6, pp. 610-622, 2000.
- [Mallet96] M. Mallet, M. Shaheen, X. Dourille and F. Chavand, « Method of matching object 3D model and its image using geometric constraints », *IEEE / IMACS Int. Conf. CESA'96, Robotics and Cybernetics*, pp. 565-569, Lille, 10-13 Juillet 1996
- [Mohan92] R. Mohan and R. Nevatia, « Perceptual organization for scene segmentation and description », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, N°12, pp. 616-632, 1999.
- [Mouaddib00] E. Mouaddib, B. Marhic, « Geometrical Matching For Mobile Robot Localisation », *IEEE Transactions On Robotics and Automation*, Vol 16, N°5, pp 542-552, October 2000.
- [Moutarlier90] P. Moutarlier and R. Chatila, « Stochastic multisensory data fusion for mobile robot location and environmental modelling », *Proc. Of the International Symposium on Robotics Research*, pp. 85-94, 1990.

- [Murray88] D.W. Murray and D. B. Cook, « Using the Orientation of Fragmentary 3D Edges Segments for Polyhedral Object Recognition », *International Journal of Computer Vision*, 2, pp. 153-169, 1988.
- [Murray88] D.W. Murray, « Strategies In Object Recognition », *The GEC Journal of research*, 6(2), pp. 80-95, 1988.
- [Neal93] R.M. Neal, « Probabilistic inference using Markov Chain Monte Carlo Methods », Technical Report CRG-TR-93-1, departement of Computer science, University of Toronto, 1993.
- [Oberkampf96] D. Oberkampf, D. F. Dementhon, L. S. Davis, « Iterative Pose Estimation using coplanar feature Points », *Computer Vision and Image Understanding*, 63(3), 495-511, 1996.
- [Pampagnin90] L.H. Pampagnin, « Reconnaissance d'objets tridimensionnels en perception monoculaire et multisensorielle », Application à la robotique spatiale », Thèse de Doctorat, Université Paul Sabatier, Toulouse, 1990.
- [Pennec98] X. Pennec, « Toward a Generic Framework for Recognition Based on Uncertain Geometric features », *Videre: Journal of Computer Vision research*, Quarterly Journal, The MIT Press, 1(2), pp. 58-88, 1998.
- [Phong95] T. Q. Phong, R. Horaud, P. D. Tao, « Object pose from 2-D to 3-D point and line correspondences », *Int. J. of Computer Vision*, Vol. 15, pp. 225-243, 1995.
- [Puget90] P. Puget, T. Skordas, « Calibrating a mobile camera », *Image and vision computing*, vol. 8, pp. 341-347, 1990.
- [Qiang89] S. Qiang, « Stratégie de localisation et identification d'objets à partir de quelques mesures tridimensionnelles », Thèse de Doctorat, Institut Polytechnique de Lorraine, 1989.
- [Quan97] L. Quan and T. Kanade, « Affine structure from line correspondences with uncalibrated affine camera », *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 19, N°8, pp. 834-845, 1997.
- [Quan99] L. Quan, Z. Lan, « Linear N-point camera pose determination », *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 21, N°8, 1999, pp. 774-780.
- [Shaheen99] M. Shaheen, « Reconnaissance d'objets polyédriques à partir d'une image vidéo pour la téléopération », Thèse de Doctorat, Université d'Evry-Val d'Essonne, 1999.
- [Shakunaga89] T. Shakunaga, H. Kaneko, « Perspective .angle Transform : Principle of Shape from Angles », *International Journal of Computer Vision*, 3, 239-254, 1989.
- [Simsarian96] K.T. Simsarian, T.J. Olson and N. Nandhakumar, « View Invariant Regions and Mobile Robot Self-Localization », *IEEE Trans. on Robotics and Automation*, 12(5), 1996, pp. 810-815.
- [Talluri93] R. Talluri and J. Aggarwal, « Position estimation techniques for autonomous mobile robot – a review », in *handbook of Pattern Recognition and computer vision*, World Scientific, Singapore, 1993, Chapter 4.4, pp 769-801.
- [Talluri96] R. Talluri and J.K. Aggarwal, « Mobile Robot Self-Location Using Model-Image feature Correspondence », *IEEE Trans. on Robotics and Automation*, Vol. 12, N°1, 1996, pp. 63-77, 1996.
- [Thrun96] S. Thrun and A. Bucken, « Integrating grid-based and topological maps for mobile robot navigation », In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 944-950, 1996
- [Ulrich00] I. Ulrich and I. Nourbakhsh, « Appearance-based place recognition for topological localization », *Proc. of the IEEE International Conference on Robotics and Automation (ICRA00)*, Vol. 2, pp. 1023-1029, 2000.
- [Wolfson90] H.J. Wolfson, « Model-based recognition by geometric hashing », In O. Faugeras, editor, *Proc. Of 1<sup>st</sup> Europ. Conf. On Comput. Vision (ECCV90)*, lecture Notes in Computer Science 427, pp. 526-536, Springer Verlag, 1990.

- [Wu94] Y. Wu, S. S. Iyengar, R. Jain, S. Bose, « A new Generalized Computational Framework for Finding Object Orientation using Perspective Trihedral Angle Constraint », IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 16, N°10, pp. 961-975, 1994.
- [Zhang00] Z. Zhang, « A flexible new technique for camera calibration », IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 22, N°11, pp. 1330-1334, 2000.
- [Zhang94] Z. Zhang, « iterative point matching for registration of free-form curves and surfaces », Int. Journal of Computer Vision, Vol. 13, N°2, pp. 119-152, 1994.