



# Perception multisensorielle de la structure géométrique d'une scène

Fano Ramparany

## ► To cite this version:

Fano Ramparany. Perception multisensorielle de la structure géométrique d'une scène. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1989. Français. NNT: . tel-00332708

**HAL Id: tel-00332708**

**<https://theses.hal.science/tel-00332708>**

Submitted on 21 Oct 2008

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

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

# THESE

*Présentée par*

Fano Ramparany

*pour obtenir le titre de* **DOCTEUR**

**de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE**

(arrêté ministériel du 5 Juillet 1984)

Spécialité: INFORMATIQUE

◆ ◆ ◆

## **Perception multisensorielle de la structure géométrique d'une scène**

◆ ◆ ◆

*Thèse soutenue le 24 Février 1989 devant la commission d'examen:*

Composition du jury:

Philippe Jorrand	Président
Nicholas Ayache Augustin Lux	rapporteurs
James L. Crowley Hugh Durrant-Whyte	examineurs



# INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : Georges LESPINARD

Année 1988

## Professeurs des Universités

BARIBAUD Michel	ENSERG	JOUBERT Jean-Claude	ENSPG
BARRAUD Alain	ENSIEG	JOURDAIN Geneviève	ENSIEG
BAUDELET Bernard	ENSPG	LACOUME Jean-Louis	ENSIEG
BEAUFILS Jean-Pierre	ENSEEG	LESIEUR Marcel	ENSHMG
BLIMAN Samuel	ENSERG	LESPINARD Georges	ENSHMG
BLOCH Daniel	ENSPG	LONGEQUEUE Jean-Pierre	ENSPG
BOIS Philippe	ENSHMG	LOUCHET François	ENSIEG
BONNETAIN Lucien	ENSEEG	MASSE Philippe	ENSIEG
BOUVARD Maurice	ENSHMG	MASSELOT Christian	ENSIEG
BRISSONNEAU Pierre	ENSIEG	MAZARE Guy	ENSIMAG
BRUNET Yves	IUFA	MOREAU René	ENSHMG
CAILLERIE Denis	ENSHMG	MORET Roger	ENSIEG
CAVAIGNAC Jean-François	ENSPG	MOSSIERE Jacques	ENSIMAG
CHARTIER Germain	ENSPG	OBLED Charles	ENSHMG
CHENEVIER Pierre	ENSERG	OZIL Patrick	ENSEEG
CHERADAME Hervé	UFR PGP	PARIAUD Jean-Charles	ENSEEG
CHOVET Alain	ENSERG	PERRET René	ENSIEG
COHEN Joseph	ENSERG	PERRET Robert	ENSIEG
COUMES André	ENSERG	PIAU Jean-Michel	ENSHMG
DARVE Félix	ENSHMG	POUPOT Christian	ENSERG
DELLA-DORA Jean	ENSIMAG	RAMEAU Jean-Jacques	ENSEEG
DEPORTES Jacques	ENSPG	RENAUD Maurice	UFR PGP
DOLMAZON Jean-Marc	ENSERG	ROBERT André	UFR PGP
DURAND Francis	ENSEEG	ROBERT François	ENSIMAG
DURAND Jean-Louis	ENSIEG	SABONNADIÈRE Jean-Claude	ENSIEG
FOGGIA Albert	ENSIEG	SAUCIER Gabrielle	ENSIMAG
FONLUPT Jean	ENSIMAG	SCHLENKER Claire	ENSPG
FOULARD Claude	ENSIEG	SCHLENKER Michel	ENSPG
GANDINI Alessandro	UFR PGP	SILVY Jacques	UFR PGP
GAUBERT Claude	ENSPG	SIRIEYS Pierre	ENSHMG
GENTIL Pierre	ENSERG	SOHM Jean-Claude	ENSEEG
GREVEN Hélène	IUFA	SOLER Jean-Louis	ENSIMAG
GUERIN Bernard	ENSERG	SOUQUET Jean-Louis	ENSEEG
GUYOT Pierre	ENSEEG	TROMPETTE Philippe	ENSHMG
IVANES Marcel	ENSIEG	VEILLON Gérard	ENSIMAG
JAUSSAUD Pierre	ENSIEG	ZADWORNÝ François	ENSERG



**Professeur Université des Sciences  
Sociales  
( Grenoble II )**

BOLLIET Louis

**Personnes ayant obtenu le diplôme  
d'HABILITATION A DIRIGER DES  
RECHERCHES**

BECKER Monique  
BINDER Zdenek  
CHASSERY Jean-Marc  
CHOLLET Jean-Pierre  
COEY John  
COLINET Catherine  
COMMAULT Christian  
CORNUEJOLS Gérard  
COULOMB Jean- Louis  
DALARD Francis  
DANES Florin  
DEROO Daniel  
DIARD Jean-Paul  
DION Jean-Michel  
DUGARD Luc  
DURAND Madeleine  
DURAND Robert  
GALERIE Alain  
GAUTHIER Jean-Paul  
GENTIL Sylviane  
GHIBAUDO Gérard  
HAMAR Sylvaine  
HAMAR Roger  
LADET Pierre  
LATOMBE Claudine  
LE GORREC Bernard  
MADAR Roland  
MULLER Jean  
NGUYEN TRONG Bernadette  
PASTUREL Alain  
PLA Fernand  
ROUGER Jean  
TCHUENTE Maurice  
VINCENT Henri

**Chercheurs du C.N.R.S  
Directeurs de recherche 1ère Classe**

CARRE René  
FRUCHART Robert  
HOPFINGER Emile  
JORRAND Philippe  
LANDAU Ioan  
VACHAUD Georges  
VERJUS Jean-Pierre

**Directeurs de recherche  
2ème Classe**

ALEMANY Antoine  
ALLIBERT Colette  
ALLIBERT Michel  
ANSARA Ibrahim  
ARMAND Michel  
BERNARD Claude  
BINDER Gilbert  
BONNET Roland  
BORNARD Guy  
CAILLET Marcel  
CALMET Jacques

COURTOIS Bernard  
DAVID René  
DRIOLE Jean  
ESCUDIER Pierre  
EUSTATHOPOULOS Nicolas  
GUELIN Pierre  
JOURD Jean-Charles  
KLEITZ Michel  
KOFMAN Walter  
KAMARINOS Georges  
LEJEUNE Gérard  
LE PROVOST Christian  
MADAR Roland  
MERMET Jean  
MICHEL Jean-Marie  
MUNIER Jacques  
PIAU Monique  
SENATEUR Jean-Pierre  
SIFAKIS Joseph  
SIMON Jean-Paul  
SUERY Michel  
TEODOSIU Christian  
VAUCLIN Michel  
WACK Bernard

**Personnalités agréées à titre  
permanent à diriger des travaux de  
recherche (décision du conseil  
scientifique)  
E.N.S.E.E.G**

CHATILLON Christian  
HAMMOU Abdelkader  
MARTIN GARIN Régina  
SARRAZIN Pierre  
SIMON Jean-Paul

**E.N.S.E.R.G**

BOREL Joseph

**E.N.S.I.E.G**

DESCHIZEAUX Pierre  
GLANGEAUD François  
PERARD Jacques  
REINISCH Raymond

**E.N.S.H.G**

ROWE Alain

**E.N.S.I.M.A.G**

COURTIN Jacques

**E.F.P.**

CHARUEL Robert

**C.E.N.G**

CADET Jean  
COEURE Philippe  
DELHAYE Jean-Marc  
DUPUY Michel  
JOUVE Hubert  
NICOLAU Yvan  
NIFENECKER Hervé  
PERROUD Paul  
PEUZIN Jean-Claude  
TAIB Maurice  
VINCENDON Marc

**Laboratoires extérieurs  
C.N.E.T**

DEVINE Rodericq  
GERBER Roland  
MERCKEL Gérard  
PAULEAU Yves

Je tiens à remercier:

Monsieur **Philippe Jorrand**, Directeur du LIFIA, qui m'a accueilli dans son laboratoire et me fait l'honneur de présider le jury de cette thèse.

Messieurs **Nicholas Ayache**, Directeur de recherche à l'INRIA,  
**Augustin Lux**, Maître de conférence à l'ENSIMAG,  
**Hugh Durrant-Whyte**, Professeur assistant à l'Université d'OXFORD,  
qui malgré leur emploi du temps chargé, ont trouvé le temps de s'intéresser à mon travail et ont accepté de participer au jury.

Monsieur **James L. Crowley**, Professeur à l'INPG, qui a su diriger mon travail avec une attention constante, tout en me laissant suffisamment d'initiative. Je le remercie également pour m'avoir orienté vers un sujet de recherche passionnant.

Je remercie Monsieur **Patrick Stelmaszyk**, Chef de projet au département Recherche d'ITMI, qui m'a permis de participer à un projet motivant.

Sont également compromis dans la rédaction de cette thèse:

**Didier Aubert** et **Christophe Discours**, pour leur complicité fréquente et quasiment indispensable.

**Philippe Caloud**, **Jocelyne Troccaz**, **Pierre Puget**, pour leur remarques succinctes, mais de qualité.

**Muriel** et **Marie**, pour leur soutien constant et généreux.

**Latex**, **MacDraw**, **Aff**, **LaserwriterNTX**, **Saturne** et la "photocopieuse du 1er", pour leur disponibilité inconditionnelle et leur patience inébranlable.

Enfin, mon père, pour m'avoir enseigné le sens de la rigueur scientifique.

A toutes ces personnes, je voudrais leur exprimer toute ma gratitude.

**Fano Ramparany**

---

Les travaux décrits dans cette thèse, ont été réalisés au LIFIA, avec la collaboration de la société ITMI.

Une partie de ces travaux a été supportée financièrement par le projet ESPRIT P940 et par le projet national ORASIS.



**"Une once de pratique vaut mieux que des tonnes  
de théorie."**

**Swâmi Sivânanda.**



## Résumé:

La précision et la richesse des informations requises par un robot pour exécuter et contrôler des tâches complexes, rend nécessaire l'utilisation conjointe de plusieurs capteurs.

Nous définissons un algorithme permettant d'exploiter la redondance et la complémentarité des informations provenant de plusieurs capteurs, pour produire, enrichir et affiner une description géométrique de l'environnement.

Cette description est basée sur une modélisation tridimensionnelles de type surfacique, qui incorpore les informations géométriques, ainsi que l'incertitude de ces informations.

Ces outils sont mis en œuvre pour réaliser un système de vision permettant de construire une description 3D d'une scène réelle, à partir d'une séquence de vues monoculaires acquises par une caméra fixée sur le bras d'un robot manipulateur.

mot-clés: perception, multi-capteurs, incertitudes, intégration, imprécision, géométrie, description, modélisation.

## abstract:

In order to perform and control complex tasks, a robot needs precise and complete information about its environment. This requirement can be satisfied using a set of multiple independant sensors.

We present a technique that exploits the redundancy and complementarity of information from multiple sensors, to build and maintain a boundary based description of the environment. Our description contains geometric information expressed as an estimate and its uncertainty.

This technique has been applied to the realisation of a vision system that produces a three-dimensional "wire frame" description of a static environment from a sequence of images taken with a camera mounted on a robot arm.

keywords: perception, multi-sensors, uncertainty, fusion, imprecision, geometry, description, modelling.



# Sommaire

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Percevoir: une nécessité . . . . .	6
1.2	Vers une perception multisensorielle . . . . .	7
1.3	L'état de l'art . . . . .	8
1.3.1	Présentation des systèmes existants . . . . .	8
1.3.2	Discussion . . . . .	18
1.4	Notre approche . . . . .	19
1.4.1	Le modèle interne: quelle représentation? . . . . .	20
1.4.2	Un algorithme d'intégration sensorielle . . . . .	21
1.5	Ce que notre travail n'aborde pas . . . . .	22
1.6	Plan du rapport . . . . .	24
<b>2</b>	<b>Modélisation de la précision</b>	<b>25</b>
2.1	Introduction . . . . .	25
2.2	Les variables aléatoires multi-dimensionnelles (v.a.m) . . . . .	26
2.3	Modélisation de la précision par les v.a.m . . . . .	31
2.3.1	Approche ensembliste . . . . .	32
2.3.2	Approche par les sous-ensembles flous . . . . .	33
2.3.3	Approche par les v.a.m. . . . .	33



2.4	Manipulation de données imprécises . . . . .	34
2.4.1	Comparaison de deux variables aléatoires multi-dimensionnelles	35
2.4.2	Combinaison de variables aléatoires multi-dimensionnelles .	36
2.4.3	Transformation de variables aléatoires multi-dimensionnelles	41
2.5	Résumé . . . . .	42
<b>3</b>	<b>modélisation géométrique</b>	<b>45</b>
3.1	Introduction . . . . .	45
3.1.1	Ce qui existe . . . . .	46
3.1.2	Nos contraintes et nos choix . . . . .	47
3.2	Primitives géométriques . . . . .	50
3.2.1	Les sommets 3D . . . . .	50
3.2.2	Les segments 3D . . . . .	51
3.2.3	Les éléments de surface . . . . .	52
3.3	Relations géométriques . . . . .	53
3.3.1	Les relations d'appartenance . . . . .	54
3.3.2	Les relations de parallélisme . . . . .	55
3.4	Structure globale de l'information géométrique . . . . .	56
3.5	Résumé . . . . .	57
<b>4</b>	<b>Représentation et gestion de l'incertitude</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.2	La perception face à l'incertitude et la contradiction . . . . .	60
4.3	L'hypothèse: une représentation de l'incertitude . . . . .	61
4.3.1	Définition d'une hypothèse . . . . .	61
4.3.2	Mesure de confiance . . . . .	62
4.3.3	Relations d'incompatibilité . . . . .	63
4.3.4	Relations de dépendance . . . . .	63
4.4	Gestion des incertitudes . . . . .	63
4.4.1	Modifications d'une hypothèse . . . . .	64
4.4.2	réévaluation d'une hypothèse . . . . .	65
4.4.3	Destruction d'une hypothèse . . . . .	68
4.5	Résumé . . . . .	70

<b>5</b>	<b>Intégration sensorielle</b>	<b>73</b>
5.1	Introduction . . . . .	73
5.2	Mise en correspondance locale de primitives . . . . .	74
5.2.1	mise en correspondance de deux segments-3D . . . . .	75
5.2.2	Mise en correspondance d'un segment-3D avec un élément de surface . . . . .	77
5.2.3	Mise en correspondance de deux éléments de surface . . . . .	77
5.3	Fusion de primitives . . . . .	78
5.3.1	Fusion de deux sommets . . . . .	78
5.3.2	Fusion de deux segments-3D . . . . .	79
5.3.3	Fusion de deux éléments de surface . . . . .	81
5.3.4	Fusion d'un élément de surface et d'un segment-3D . . . . .	83
5.4	Stratégie de mise en correspondance globale . . . . .	84
5.5	Algorithme d'un cycle d'intégration . . . . .	87
5.5.1	Principes . . . . .	87
5.5.2	historique d'un cycle d'intégration . . . . .	89
5.6	Résumé . . . . .	92
<b>6</b>	<b>Application: la caméra mobile</b>	<b>93</b>
6.1	Principe . . . . .	94
6.2	Architecture matérielle . . . . .	94
6.3	Obtention des données tri-dimensionnelles . . . . .	95
6.3.1	Calibrage du système . . . . .	96
6.3.2	Traitement des images . . . . .	98
6.3.3	Technique de la projection inverse . . . . .	100
6.3.4	Expérimentation . . . . .	101
6.4	Résultats expérimentaux . . . . .	105
6.4.1	Remarque préliminaire . . . . .	105
6.4.2	Initialisation et amorçage du processus d'intégration . . . . .	107
6.4.3	Evolution du modèle . . . . .	109
6.4.4	Autres résultats . . . . .	112
6.5	Discussion . . . . .	113
6.6	Résumé . . . . .	114

---

<b>7</b>	<b>Application: La stéréoscopie trinoculaire</b>	<b>119</b>
7.1	Principe . . . . .	119
7.2	Conditions de l'expérience . . . . .	120
7.3	Construction du modèle . . . . .	120
7.3.1	Traitement des images . . . . .	120
7.3.2	Intégration des trois vues . . . . .	122
7.4	Discussion des résultats . . . . .	124
<b>8</b>	<b>Conclusion</b>	<b>129</b>
8.1	L'évaluation de notre approche . . . . .	129
8.2	Perspectives futures . . . . .	130
8.2.1	Les autres capteurs? . . . . .	130
8.2.2	Le modèle composite: pour quoi faire? . . . . .	131
8.2.3	Vers une description plus riche . . . . .	131
8.2.4	Un univers dynamique . . . . .	131
<b>A</b>	<b>Calibrage de la caméra</b>	<b>143</b>
<b>B</b>	<b>Calibrage du lien robot-caméra</b>	<b>147</b>
<b>C</b>	<b>Technique de la projection inverse</b>	<b>151</b>

# Chapitre 1

## Introduction

Les travaux présentés dans ce rapport, traitent de la perception de la géométrie de l'environnement d'un robot mobile ou d'un robot manipulateur. Nous y avons abordé deux points:

- La représentation des informations géométriques.
- La définition d'un mécanisme d'intégration des informations, lorsque le robot utilise plusieurs capteurs.

Dans ce chapitre, nous définissons la perception et évaluons son enjeu dans le contexte de la robotique, puis nous justifions l'intérêt d'une approche multi-capteurs. Nous analysons ensuite les systèmes existants, en dégagant leurs apports et leurs points faibles. Nous introduisons enfin notre approche dans ses grandes lignes, et présentons le plan du rapport.

## 1.1 Percevoir: une nécessité

L'exécution et le contrôle de tâches complexes telles que la reconnaissance d'objets, la manipulation d'objets, la navigation, l'évitement d'obstacles, s'avèrent difficiles dans un environnement complexe et non structuré, du fait du manque d'informations et de l'imprécision de l'information disponible. Pour réaliser correctement ces tâches, le robot doit disposer d'informations sur la géométrie de l'environnement, en qualité et en quantité suffisante. Pour saisir correctement un objet, le robot doit en connaître la position précise sur la table, ainsi que l'espace occupé par les objets environnants. La connaissance de ces informations, permettra au robot d'éviter les collisions lors de la phase d'approche de l'objet et donc d'atteindre son objectif.

Par information géométrique, on entend la donnée des entités géométriques qui composent l'environnement. Les surfaces, les arêtes, les coins des objets sont des exemples d'entités géométriques. En particulier, il est utile de disposer des caractéristiques quantitatives sur ces entités, telles que la position, l'orientation et les dimensions des surfaces planes. Une représentation de ces informations est appelée **modèle géométrique de la scène**.

Dans un environnement parfaitement maîtrisé, comme celui d'une cellule d'assemblage, la géométrie et la position des pièces à manipuler sont connues à l'avance. Ces informations sont "entrées à la main" par l'opérateur, ou interviennent directement dans le "code" source du programme de commande du robot. Par contre, en milieu inconnu, incertain ou imprévu, le robot doit disposer d'une source lui fournissant ces informations. La perception joue ce rôle. Nous en proposons la définition suivante:

*La Perception est la fonction qui permet à un agent <sup>1</sup> de construire une description interne de son environnement physique, à partir des informations que lui fournissent ses organes sensoriels. Cette description interne est aussi appelée **modèle de l'environnement**.*

Le modèle de l'environnement contient alors l'information géométrique nécessaire

---

<sup>1</sup>Nous appelons agent une entité douée d'autonomie, et qui possède la faculté d'agir sur son environnement.

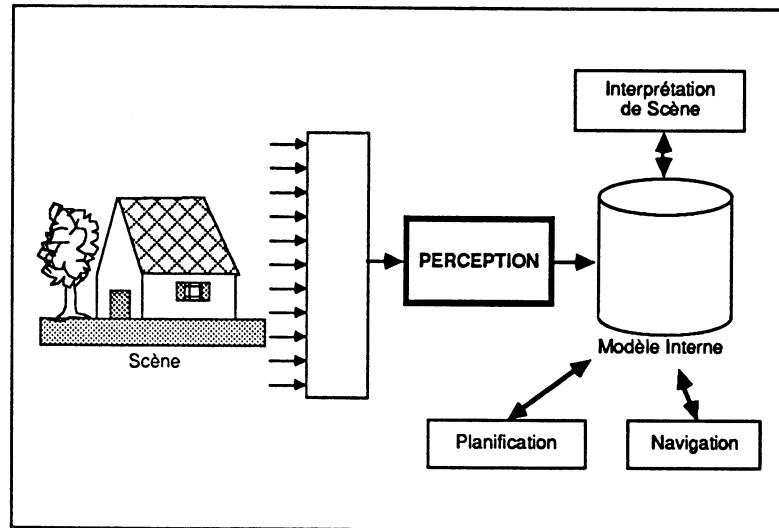


Figure 1.1: Le rôle de la perception: construire le modèle interne de l'environnement

aux modules de raisonnement, tels que la planification d'actions, l'interprétation de scène. Par la suite, nous appellerons ces modules, modules amonts. figure 1.1.

la perception est une fonction fondamentale dans la conception du robot de la "prochaine génération", car elle permettra de passer de l'ère des robots "aveugles" agissant sur un univers figé et sans "surprises", à l'ère des robots "intelligents", possédant la faculté de s'adapter à un univers imprévisible, ou inconnu. La perception constitue donc un point clé, dans l'accession des robots à plus d'autonomie.

## 1.2 Vers une perception multisensorielle

Dans le contexte de la robotique autonome, les données sensorielles brutes sont fournies par des capteurs. Il existe sur le marché une grande diversité de capteurs. Parmi les plus couramment utilisés, on peut citer les systèmes stéréoscopiques, les systèmes de vision 3D, les capteurs ultrasoniques, les capteurs tactiles.

Chacun de ces capteurs pris isolément, fournit des informations qui sont incomplètes et généralement insuffisantes, pour permettre la réalisation des tâches robotiques complexes. Par exemple, les capteurs tactiles ne fournissent qu'une information locale, sur les surfaces à portée du robot. Cette technique est insuffisante pour

reconnaître et localiser les objets éloignés. Inversement, un système stéréoscopique fournit des données trop imprécises pour envisager une manipulation minutieuse.

Ce constat met en évidence l'intérêt de faire coopérer plusieurs capteurs, pour pallier à leur insuffisance respective, et exploiter la complémentarité des informations qu'ils fournissent. Les mécanismes mis en œuvre pour faire coopérer différents capteurs ont donné jour à la notion d'intégration multisensorielle.

**L'intégration ou fusion de données multisensorielles** est le processus qui consiste à combiner les informations provenant de plusieurs capteurs, ou de différents types de capteurs. Cette combinaison a pour but d'accroître la précision des données contenues dans le modèle, de résoudre les ambiguïtés qui y sont présentes, et d'enrichir le modèle.

Dans le paragraphe suivant, nous présentons l'état de l'art, dans le domaine de l'intégration de données multisensorielles.

### 1.3 L'état de l'art

#### 1.3.1 Présentation des systèmes existants

Nous passons en revue les systèmes de vision les plus connus, en esquisant leurs grandes lignes, puis en détaillant leur capacité d'intégration sensorielle.

##### **Mosaic-3D**

[HK86]

Mosaic-3D est un système de vision qui permet de traiter une séquence de paires d'images stéréoscopiques, ou d'images monoculaires, pour construire incrémentalement une description tri-dimensionnelle de la scène.

Mosaic-3D fonctionne selon le principe suivant: chaque nouvelle acquisition de données fournit soit une paire d'images, soit une image simple. Un module d'analyse stéréoscopique permet de traiter les paires d'images. Un module d'analyse monoculaire permet de traiter les images isolées. Chacun de ces deux traitements fournit au système, une description tri-dimensionnelle de la scène sous forme d'un ensemble

de jonctions 3D. Un module de construction et de mise à jour permet d'intégrer chaque description pour enrichir un modèle courant de la scène. Le modèle de la scène est également tri-dimensionnel, et permet de représenter les sommets, les facettes planes, les groupes d'arêtes qui relient les côtés des facettes planes, les plans, les droites et les points.

L'intégration de nouvelles données se traduit par une modification des données du modèle, qui tient compte des nouvelles données acquises, ainsi que par l'addition de faces occultées dans les images précédentes.

L'utilisation du système est illustrée par la reconstruction d'une scène urbaine, à partir de plusieurs vues aériennes.

Nous pouvons noter la puissance de représentation du système, qui s'inspire des concepts introduits par la C.A.O., à savoir une modélisation géométrique, organisée en un "graphe de structure" dont les noeuds représentent des primitives géométriques, et les arcs des contraintes topologiques et géométriques.

Des noeuds hypothétiques sont créés, pour pallier à l'incomplétude du modèle de la scène. Ces noeuds hypothétiques sont générés par des règles définissant les objets à faces planes, telles que la connexité des arêtes délimitant une facette plane, la définition d'une arête par les sommets situés à ses extrémités, l'appartenance de toute arête à deux facettes planes adjacentes. Ce mécanisme de complétement du modèle à l'aide de primitives hypothétiques, permet de réaliser la mise en correspondance entre les nouvelles données et le modèle courant à un niveau objet. Ce qui permet de regrouper les primitives, et donc de diminuer la complexité de l'opération de mise en correspondance.

Néanmoins, Mosaic-3D n'est pas modifiable ou extensible pour une application à d'autres types de scènes, car les connaissances qui étaient utilisées pour compléter la description de la scène à l'aide de primitives hypothétiques, qui sont propres aux objets polyédriques, sont intégrées au système lui-même.

De plus, la précision des mesures n'est pas prise en compte explicitement lors de la phase de combinaison des données. Les auteurs attribuent les différences entre données acquises et données du modèle, aux incertitudes respectives de ces



données. Leur combinaison est réalisée à l'aide d'une simple moyenne arithmétique, en pondérant différemment les nouvelles données et celles du modèle. Nous pensons qu'une telle approche empirique nuit à la robustesse du système.

### 3-D FORM

[WHK88]

Le système 3-D FORM est le successeur du système Mosaic-3D, présenté précédemment. Il possède les mêmes fonctionnalités que Mosaic-3D, et remédie à certaines insuffisances de celui-ci. En particulier, l'utilisation des "schémas" (ou frames) pour représenter les primitives et les relations géométriques, a permis de gagner en souplesse de représentation. A l'aide de ce mode d'implémentation, il est aisé d'ajouter de nouvelles primitives. Si la nouvelle primitive est une classe particulière d'une primitive plus générale qui existait déjà, le mécanisme d'héritage, permet de ne spécifier que les différences ou les particularités propres à la nouvelle primitive.

L'utilisation de schémas permet un contrôle plus efficace pour deux raisons:

- Le contrôle peut être hiérarchique, grâce au caractère lui-même hiérarchique de la représentation. En effet, on peut mettre en correspondance les objets les plus généraux d'abord, les objets les plus particuliers ensuite. Cette stratégie permet en cas d'échec de mise en correspondance à un niveau hiérarchique donné, d'éviter la mise en correspondance de toutes les primitives plus particulières.
- Une forme particulière du "focus d'attention" a pu être implémentée naturellement, à l'aide de démons attachés aux primitives. En effet, l'accès à une primitive permet d'activer un démon qui détermine ce que l'on doit faire ensuite, et où le faire. On s'affranchit ainsi d'un processus de contrôle externe, qu'il est difficile de maintenir lorsque le système évolue.

La figure 1.2 représente le type de données résultant d'une analyse monoculaire, le type des objets génériques dont on veut trouver des instances dans les données vues, et la description géométrique des immeubles résultant de la reconstruction.

Le système 3D-FORM diffère donc de son prédécesseur Mosaic-3D, par son implé-

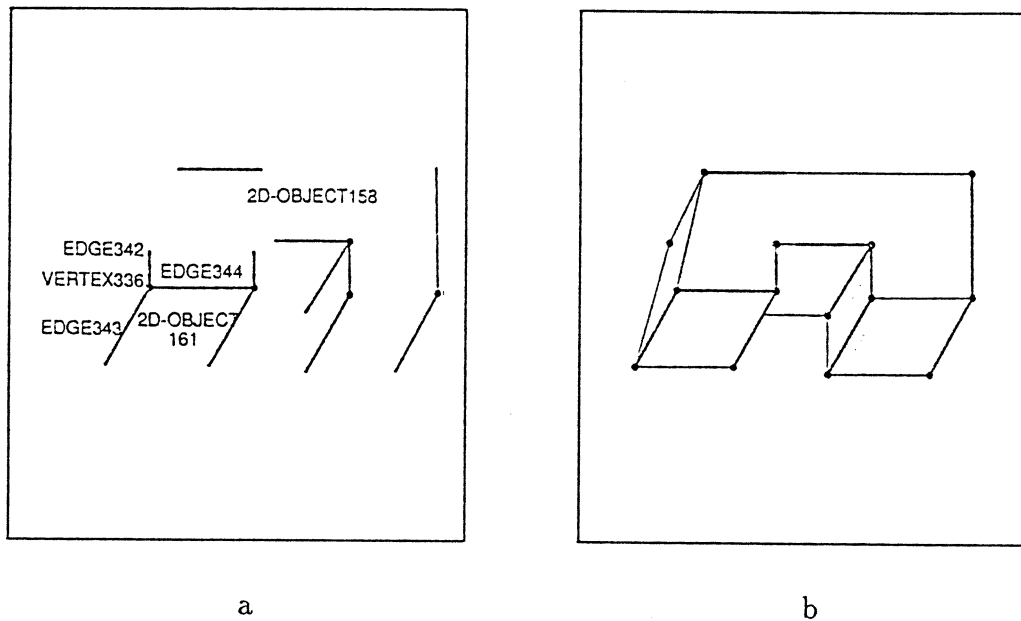


Figure 1.2: Types de données manipulées par 3D-FORM [E.L.Walker & al. A framework for representing and reasoning about three-dimensional objects for vision. AI Magazine n.9, 1988]

mentation à l'aide des schémas. L'utilisation de cet outil d'implémentation, confère au système une certaine souplesse dans la définition de nouvelles entités géométriques, ainsi que dans la mise en œuvre du contrôle du système.

Cependant, 3D-FORM ne possède toujours pas de traitement des mises en correspondance erronées. Par exemple, si deux classes ont été regroupées par erreur, et que l'objet résultant ne correspond à aucun modèle, il n'y a pas d'exploration des autres alternatives à la mise en correspondance erronée.

### Certainty grids (Grilles de certitude)

[Mor88]

Hans Moravec introduit une représentation de l'espace libre d'un robot mobile, à l'aide d'une grille 2D, appelée "grille de certitude", qui discrétise l'environnement dans lequel le robot est susceptible de se déplacer (figure 1.3).

Chaque cellule de la grille délimite une portion carrée de l'environnement. A

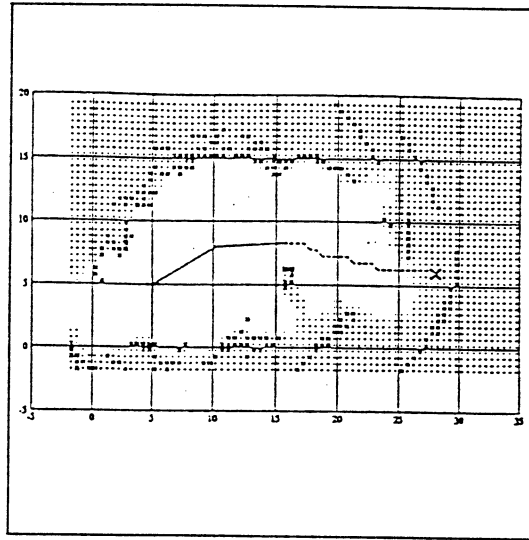


Figure 1.3: Grille de certitude [H.P.Moravec, Sensor fusion in certainty grids for mobile robots. AI Magazine n.9, 1988]

chaque cellule est affectée la valeur d'un "facteur d'occupation"  $L(o)$ , qui donne une indication sur l'occupation de la cellule.

- Lorsque  $L(o) = 0$ , la cellule est vide, et fait donc partie de l'espace libre.
- Lorsque  $L(o) = \infty$ , la cellule est occupée, et fait donc partie d'un obstacle physique.
- Lorsque  $0 < L(o)$ , l'occupation de la cellule est incertaine. Néanmoins, cette incertitude est nuancée. Lorsque  $L(o)$  est proche de 0, la cellule est vraisemblablement vide, lorsque  $L(o)$  est très grand, la cellule est vraisemblablement occupée.

$L(o)$  est défini par la formule suivante:

$$L(o) = \log\left(\frac{p(o)}{p(\neg o)}\right) \quad (1.1)$$

où  $p(o)$  est la probabilité que la cellule soit occupée, et  $p(\neg o)$  la probabilité que la cellule soit libre.

Le robot dispose de capteurs à ultra-son, d'un système stéréo 2D et de capteurs de proximité pour explorer la géométrie de son environnement. A un instant donné,

chaque capteur fournit une instance de la “grille de certitude”, en affectant à chaque cellule, une opinion sur son occupation. Cette opinion est exprimée sous la forme d’une probabilité conditionnelle  $p(o|A)$ , qui caractérise l’évènement  $o = \text{“la cellule est occupée”}$ , sur la base de mesures fournies par le capteur  $A$ . Les instances de la grille de certitude fournies par chacun des capteurs sont fusionnées pour ne former qu’une seule grille dont chaque cellule contient un consensus des opinions de chacun des capteurs, basé sur la formule de Bayes.

Dans le cas de deux capteurs  $A$  et  $B$ , si  $p(o|A)$  représente l’opinion du capteur  $A$  sur l’occupation d’une cellule de la grille, et  $p(o|B)$  celle du capteur  $B$ , la fusion des deux opinions s’effectue suivant la formule:

$$p(o|A \wedge B) = \frac{p(B|o \wedge A) \times p(o|A)}{p(B|o \wedge A) \times p(o|A) + p(B|\neg o \wedge A) \times p(\neg o|A)} \quad (1.2)$$

Dans le cas où les capteurs sont indépendants, les auteurs montrent que la formule 1.2 conduit à l’expression suivante:

$$L(o|A \wedge B) = L(o|A) + L(o|B) - L(o) \quad (1.3)$$

qui remet à jour le facteur d’occupation  $L(o|A \wedge B)$ , connaissant les mesures  $L(o|A)$  et  $L(o|B)$  des capteurs  $A$  et  $B$ , ainsi que la valeur initiale  $L(o)$ .

Par ces travaux, Hans Moravec met en évidence l’intérêt d’utiliser une représentation uniforme, pour exprimer les informations sensorielles. Néanmoins, si la description bi-dimensionnelle basée sur un échantillonnage de l’environnement suffit à la tâche de navigation, elle s’avère inadaptée pour des tâches plus complexes.

### Le système de vision de l’INRIA

[AF88]

Le système de vision de l’INRIA<sup>2</sup> est basé sur l’utilisation d’un triplet de caméras. Les trois caméras sont fixées de façon rigide à la plateforme d’un robot-moblie. Le robot-moblie se déplace dans son environnement en acquérant à différentes places de sa trajectoire, un triplet d’images. Chaque triplet d’images est analysé par

---

<sup>2</sup>INRIA: Institut National de Recherche en Informatique et Automatique

stéréoscopie. Cette analyse fournit une description tri-dimensionnelle de type “fil de fer”, rapportée à un repère attaché au triplet de caméras.

Ainsi, après  $n$  prises d’images effectuées en  $n$  positions différentes du robot, le système dispose de  $n$  descriptions de type “fil de fer” de la scène. Le traitement de ces descriptions se fait alors en deux étapes:

**L’appariement:** deux descriptions successives sont appariées à l’aide du module HYPER-3D. HYPER-3D permet l’appariement des indices tri-dimensionnels deux-à-deux, mais fournit aussi une estimation du déplacement entre les deux positions du triplet.

**La Fusion:** les données des indices appariés sont combinées à l’aide de la technique du Filtre de Kalman. Cette opération permet de réduire l’incertitude de l’indice résultant de la combinaison.

Deux points abordés dans les travaux de l’INRIA, nous semblent importants.

Le premier, concerne la représentation explicite des incertitudes à l’aide de modèles gaussiens. Ce mode de représentation présente plusieurs avantages:

- La description est compacte. L’incertitude d’un point, de coordonnées  $(x, y)$  dans un plan, est représentée par une simple matrice symétrique  $2 \times 2$ , donc 3 paramètres.
- La description est souple. Les auteurs utilisent un modèle unique, pour décrire l’incertitude des données géométriques et l’incertitude de position des capteurs.
- Les calculs de combinaison des incertitudes sont très simples et sont réalisés à l’aide de la technique du filtre de Kalman.

La seconde notion importante est celle d’oubli. L’oubli consiste à ne pas mémoriser des informations redondantes. Par exemple, certains segments sont susceptibles d’être détectés par le système stéréoscopique. Si  $S$  est un tel segment, il va produire un segment  $S_1$  dans la position 1, un segment  $S_2$  dans la position 2,... un segment

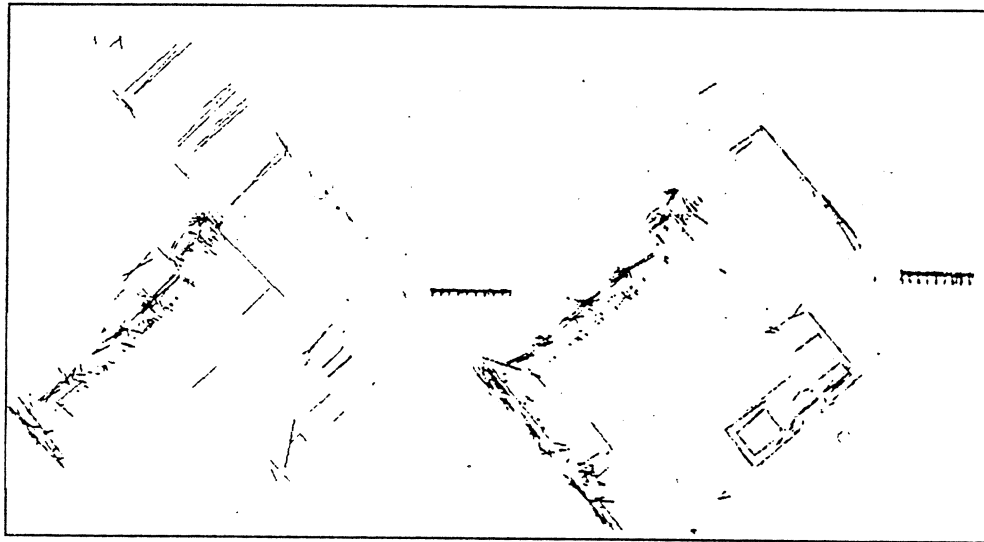


Figure 1.4: Types de données manipulées par le système de l'INRIA [N.Ayache & al. Maintaining representations of the environment of a mobile robot. Technical Report 789, INRIA Rocquencourt, 1988]

$S_n$  dans la position  $n$ . Une fois acquis ces différents segments, et une fois ceux-ci combinés, le système peut se permettre d'oublier  $S_1, S_2, \dots, S_n$  pour ne retenir que  $S$  qui incorpore toutes les informations utiles extraites des segments  $S_1, S_2, \dots, S_n$ . La figure 1.4 représente une "carte visuelle" contenant les segments 3D produits par une analyse trinoculaire, et les segments 3D résultant de la fusion de plusieurs "cartes visuelles".

Cependant, la représentation géométrique reste très sommaire, de l'avis même des auteurs. Seules sont décrites les discontinuités de surfaces, telles que les arêtes d'objets polyédriques. Pour produire une description plus riche, incluant une représentation des surfaces, l'INRIA propose une technique permettant de polygonaliser l'espace libre, en appuyant les facettes du polygone, sur les segments 3D fournis par le système stéréoscopique.

### Le système d'intégration de l'université de Pennsylvanie

[Dur86]

Hugh F. Durrant-Whyte, présente une méthode permettant d'intégrer des mesures

éparses d'indices géométriques produites par différents capteurs, en une modélisation de l'état de l'environnement. Il traite essentiellement deux problèmes:

- La représentation topologique de l'environnement, qui se situe à deux niveaux.
  - un niveau supérieur “objet”, qui permet de décrire les objets physiques de la scène. Chaque objet possède un repère qui lui est propre, et qui le localise dans la scène. Les objets sont reliés entre eux par des relations topologiques invariantes par changement de repère. L'*objet*<sub>1</sub> est “posé sur” l'*objet*<sub>2</sub> est un exemple de relation. Les contraintes de stabilité imposent des contraintes sur l'orientation relative de l'*objet*<sub>1</sub> par rapport à l'*objet*<sub>2</sub>.
  - un niveau inférieur “indice géométrique”, qui permet de modéliser la géométrie des objets.

Les indices géométriques sont affectés d'une incertitude, que l'auteur modélise par une distribution “gaussienne contaminée”. Cette mesure permet de décrire un phénomène possédant une dispersion moyenne de distribution gaussienne  $Gauss(m_1, \Lambda_1)$ , perturbée par des écarts importants et rares, possédant une distribution  $Gauss(m_2, \Lambda_2)$ , et qui permettent de modéliser les valeurs aberrantes du phénomène. La distribution gaussienne contaminée s'écrit alors:

$$ContaminatedGauss = (1 - \epsilon)Gauss(m_1, \Lambda_1) + \epsilon Gauss(m_2, \Lambda_2) \quad (1.4)$$

$\epsilon$  définit la fréquence relative d'occurrence des écarts importants, par rapport à celle des écarts moyens.

- Le processus d'intégration. Celui-ci se déroule en trois temps:
  1. La comparaison des mesures des différents capteurs permet de regrouper celles-ci, en éliminant celles qui sont aberrantes,
  2. Leur combinaison met à jour l'objet du modèle, auquel ces mesures se rapportent.
  3. La propagation de l'information, s'effectue par le biais du réseau de relations topologiques. Cette propagation modifie les objets connexes à celui remis à jour, afin que soient préservées les relations invariantes du réseau.

Durrant-Whyte réalise une bonne analyse de l'application des distributions gaussiennes à la modélisation de l'incertitude. En particulier, il étudie les conséquences liées à la dépendance des mesures, sur leur combinaison.

Du fait de l'algorithme du processus d'intégration, le système ne peut s'appliquer que sur un environnement connu, dans le sens où les objets et les relations entre les objets sont connus, mais incertain, dans le sens où la localisation des objets peut être approximative. Aussi, lors des expérimentations, le réseau est d'abord stocké manuellement, puis le système réactualise les positions des objets, en fonction de mesures que lui fournissent les capteurs.

### **Les autres systèmes**

Il existe d'autres travaux portant sur la fusion de données multisensorielles. Ceux-ci n'abordent pas le problème de fusion dans sa généralité, mais traitent de points spécifiques, ou sont menés dans le cadre d'une application très particulière.

Shekhar [SKS86] étudie le problème de localisation d'objet dans un environnement multisensoriel.

Mitiche [MA86] combine les informations contenues dans différents types d'images: des images d'intensité, des images de profondeur et des images infra-rouges.

Kamat [Kam85] situe le processus d'intégration à un niveau de description objet. L'environnement contient un ensemble fini d'objets. Chaque capteur est défini par un ensemble de règles de décision qui identifient les objets à partir d'indices observés. Un démonstrateur de théorème permet de rejeter les candidats potentiels en appliquant les règles propres aux capteurs.

Alspach [Als83] présente le problème de fusion de données et discute les différentes alternatives au choix d'une architecture d'implémentation d'un module d'intégration multisensorielle. Notamment, Alspach étudie l'influence du nombre de capteurs, et l'organisation des flots de données sur le coût de calcul du processus d'intégration. On peut citer les travaux de Porrill ([Por87]), sur l'effet de contraintes géométriques sur la fusion de points et de droites 3D. Ces contraintes comprennent l'orthogonalité, l'intersection, l'identité de primitives géométriques. Elles peuvent être hypothétiques, puis testées après fusion des primitives, pour en vérifier la validité.



Dans les articles cités jusqu'ici, le processus de perception doit traiter la redondance d'information produite par la multiplicité des sources d'information. D'autres travaux prônent une approche basée sur la complémentarité des sources d'informations ([SST86],[RdSV86]). Cette approche est plus efficace car elle évite le problème délicat de mise en correspondance des informations, et définit la stratégie de perception par les besoins et par la connaissance des performances des capteurs. Mais comme toute approche utilisant des informations externes au module de perception, elle s'avère trop spécifique, pour fournir un cadre général d'intégration multi-sensorielle.

Enfin, on ne peut pas ne pas citer les idées présentées par J.L.Crowley ([Cro85]), dont nos travaux se sont largement inspirés. Les principes de construction et maintenance d'un modèle surfacique de l'environnement sont introduits dans [Cro85]. Ceux-ci ont été appliqués dans le cas bi-dimensionnel, pour modéliser l'environnement d'un robot mobile muni de capteurs ultra-soniques ([Cro86]).

### 1.3.2 Discussion

De notre étude bibliographique, nous avons constaté que très peu de systèmes de vision abordent le problème de l'intégration multi-sensorielle, ou de mise à jour dynamique d'informations sensorielles.

De plus, les représentations des informations géométriques utilisées par ces systèmes, sont généralement trop pauvres pour constituer des bases de données solides, à partir desquelles des tâches robotiques complexes peuvent être envisagées. Seul, 3D-FORM [WHK88] envisage un jeu de primitives comprenant des jonctions, des arêtes et des facettes planes. Dans les autres systèmes l'environnement est soit bi-dimensionnel, soit tri-dimensionnel, mais alors le jeu des primitives géométriques utilisées est limité.

Quelques systèmes adoptent l'hypothèse d'erreurs possédant une distribution gaussienne, pour modéliser l'incertitude ou précision numérique des primitives géométriques. Malheureusement, aucun auteur ne discute du fondement théorique de cette modélisation.

Enfin, aucun système ne propose de traiter le cas où les informations sensorielles seraient contradictoires, soit du fait de données sensorielles brutes inexactes, soit du fait d'inférences intermédiaires fausses. Or, il nous semble indispensable que le système puisse traiter le cas de données incertaines, qui bien que rendant temporairement incohérent l'état courant du modèle de l'environnement, peuvent véhiculer des informations importantes. En effet, les ignorer à un instant donné peut priver le modèle de la scène de renseignements qui s'avèreront nécessaires par la suite.

## 1.4 Notre approche

Rappelons les objectifs de nos travaux.

Notre ambition est d'élaborer un système de perception pour la robotique autonome, dans un contexte multisensoriel. Le système visé doit être capable de construire une représentation interne de la géométrie de l'environnement, à partir des informations fournies par plusieurs capteurs.

Notre premier choix consiste à imposer l'utilisation d'une description intermédiaire permanente entre les mesures brutes des capteurs et les modules amonts. Cette description intermédiaire est nécessaire car le niveau de description des mesures en sortie des capteurs, ne correspond jamais au niveau de description requis par les modules amonts. Ce problème est traité par D.Marr [Mar82] et Y.Demazeau [Dem86] dans le domaine de la vision par ordinateur. Le caractère permanent de cette description peut être vu comme une "mémoire à moyen terme". Elle évite l'usage abusif des capteurs en privilégiant l'utilisation des informations contenues dans une structure permanente. En effet, dans l'état actuel des techniques, le coût de traitement des signaux de capteurs est supérieur à celui de la recherche d'information dans une base de donnée correctement structurée.

Le second choix porte sur le niveau de description: Doit-on utiliser une description de type iconique, c'est-à dire organisée sur un plan, ou une description purement symbolique? Le choix d'une représentation géométrique est argumenté dans la section suivante.

### 1.4.1 Le modèle interne: quelle représentation?

Le monde réel est organisé par la géométrie. Plus précisément, la description que nous nous en faisons est essentiellement géométrique. Lorsque nous décrivons une scène, nous nous exprimons généralement en des termes géométriques. Par exemple: *“le ballon est posé sur la table”, “un ballon est sphérique”, “une table possède une surface plane”*. De plus la géométrie est un langage universel. Elle permet de recouvrir un large nombre de capteurs. Pour combiner les données provenant de plusieurs capteurs, il est pratique de les exprimer dans un même formalisme, et dans des termes qui permettent de les comparer. Ceci nous a amené tout naturellement à exprimer la représentation interne en des termes géométriques.

Nous abordons maintenant le choix de la représentation des informations géométriques. C'est une étape importante de notre recherche. En effet, le mode de représentation que nous adopterons, conditionnera fortement les processus qui vont utiliser ou traiter ces informations. Notamment, lors de la confrontation entre les informations contenues dans la représentation interne et les informations provenant des capteurs, le système devra mettre en œuvre des stratégies de mise en correspondance. L'expérience montre, que les opérations de mise en correspondance sont très coûteuses en place mémoire et en temps d'exécution, et qu'elles constituent généralement le goulot d'étranglement des processus qui les mettent en œuvre. La complexité de ces opérations dépend directement du mode de représentation des données, et particulièrement de la façon dont ces données sont organisées.

Le problème du choix de la représentation, nous amène naturellement à répondre à deux questions :

1. quel usage doit-on faire de cette représentation ?
2. qu'est-il utile ou nécessaire de représenter ?

Nos réponses à ces questions nous imposent des contraintes sur le mode de représentation à adopter. Ces contraintes nous guideront vers le mode le plus approprié.

Pour réaliser des tâches complexes, il faut des données précises et non-ambiguës

de l'environnement. Cette première contrainte, élimine d'ores et déjà les représentations bi-dimensionnelles, et incite à s'orienter plutôt vers une représentation tri-dimensionnelle. Les travaux menés en robotique d'assemblage, montrent qu'il est nécessaire de représenter les entités géométriques telles que les coins, les arêtes, les surfaces des objets. Le mode de représentation doit décrire ces entités.

Les capteurs se différencient par la précision des données qu'ils fournissent. Pour réaliser des montages fins et précis, le robot doit tenir compte non seulement de l'information sur la position et l'orientation des objets qu'il manipule, mais aussi de la précision de cette information. Nous sommes donc amenés à représenter explicitement la précision des données. Ceci offre à notre avis, un caractère de généralité au mode de représentation. Par ce biais, il sera plus aisé d'exprimer les informations à l'aide d'un mode de représentation uniforme, même si celles-ci proviennent de capteurs différents.

Les capteurs se différencient par leur fiabilité. Certains capteurs fournissent des données auxquelles le robot peut se fier. Par exemple, un capteur tactile permet de détecter la présence et la proximité d'une surface, sans aucune ambiguïté sur l'interprétation des mesures qu'il fournit. Par contre, certains capteurs fournissent des informations incertaines. Par exemple, un système stéréoscopique peut produire des indices géométriques tri-dimensionnels erronés, par suite d'une mise en correspondance inexacte. D'une façon générale, l'imprécision des données sensorielles peut fausser leur interprétation, et donc générer un modèle incorrect. Il est important que le système puisse prendre en compte, aussi bien les données sûres, que celles qui sont incertaines. Ainsi, la description de l'environnement pourra-t-elle être riche et dense. Le caractère incertain des informations contenues dans cette description, pourra être atténué par un mécanisme permettant de conforter les données incertaines, à l'aide des observations ultérieures. Pour cette raison, il nous paraît utile de représenter explicitement la fiabilité des informations.

#### 1.4.2 Un algorithme d'intégration sensorielle

Le deuxième problème, est celui de définir une méthode, permettant de combiner les informations fournies par les différents capteurs, pour produire la représentation

interne de l'environnement.

Ici encore, notre premier souci est d'offrir à notre méthode un caractère de généralité, afin que celle-ci soit indépendante du type des capteurs utilisés, ainsi que de leur nombre. Il nous semble que la solution la plus appropriée à satisfaire cette exigence, consiste à introduire un niveau de description intermédiaire, permettant d'exprimer les informations brutes fournies par les capteurs, à l'aide d'un mode de représentation unique.

Dans le but de faciliter le processus d'intégration, il est pratique d'utiliser le même mode de représentation, pour décrire la description intermédiaire et la modélisation interne de l'environnement. Cette approche permet d'envisager le mécanisme d'intégration, comme un processus cyclique, où les informations brutes des capteurs sont traitées en deux étapes. Lors de la première étape, les informations sont exprimées à l'aide du même mode de représentation. Chaque capteur fournit donc une description partielle de l'environnement. La seconde étape, permet d'intégrer la description au modèle de l'environnement.

Initialement, le modèle peut être vide. Ensuite, les différentes descriptions sont utilisées successivement pour mettre à jour le contenu de la description interne (voir figure 1.5). Le processus s'arrête lorsque toutes les descriptions ont été utilisées.

Notre approche possède ainsi l'avantage d'être transparente au type de capteur utilisé, pourvu que l'on puisse décrire les informations qu'il fournit, à l'aide du mode de représentation uniforme que nous avons défini.

Nous nous sommes penchés sur les différents problèmes que nous venons d'évoquer. Leur analyse et les solutions que nous proposons font l'objet du rapport.

## 1.5 Ce que notre travail n'aborde pas

Nos hypothèses de travail supposent que les capteurs effectuent leurs acquisition simultanément, et que l'information sur la scène est relative à l'instant d'acquisition. Pratiquement, cette condition est difficilement réalisable, mais il est possible de s'en affranchir lorsque la scène est statique. Par exemple lors des expérience que nous

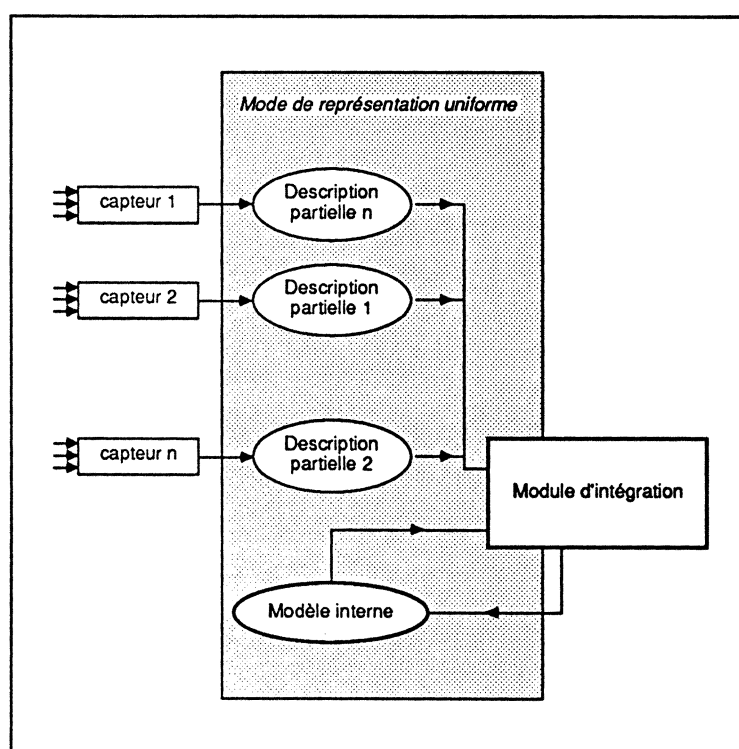


Figure 1.5: schéma fonctionnel du processus d'intégration

décrivons au chapitre 6, nous effectuons des acquisitions de données avec un capteur unique, en des instants successifs. Le caractère statique de la scène nous permet de traiter ce problème comme une fusion de données acquises simultanément par plusieurs capteurs.

Nous n'avons pas abordé le problème de la dynamique de la scène. Pour tenir compte de cet aspect, on pourra dans un premier temps reconstruire la description interne à chaque fois que la scène subit des modifications.

Le paragraphe suivant, présente l'articulation du rapport.

## 1.6 Plan du rapport

Dans le chapitre 2, nous nous intéressons à la représentation de la précision des données.

Dans le chapitre 3, nous définissons un mode de représentation uniforme des informations géométriques.

Puis nous développons, dans le chapitre 4, un formalisme qui nous permet de représenter et gérer l'incertitude de l'information, contenue dans la description interne de l'environnement.

Dans le chapitre 5, ces différents outils sont utilisés pour définir le mécanisme d'intégration, qui permet de mettre à jour la description interne, en fonction des informations sensorielles.

Le principe est alors appliqué à deux problèmes pratiques.

1. La reconstruction de scène, à partir d'une caméra mobile. (chapitre 6)
2. La stéréoscopie trinoculaire. (chapitre 7)

La description des expériences et résultats obtenus sont présentés et discutés dans le chapitre 6.

## Chapitre 2

# Modélisation de la précision

### 2.1 Introduction

Dans ce chapitre, nous nous intéressons à la modélisation de la précision des données numériques.

La notion de précision des informations est fondamentale dans un cadre multi-sensoriel. En effet, certains capteurs fournissent des données très précises: les télémètres lasers, les capteurs tactiles mesurent des profondeurs avec une précision supérieure au millimètre. D'autres capteurs fournissent des données moins précises: les capteurs ultra-son, les systèmes stéréoscopiques mesurent les profondeurs avec des erreurs, qui peuvent atteindre le centimètre pour des objets éloignés. La fusion d'informations hétérogènes provenant de capteurs différents, nécessite de prendre en considération les précisions relatives des capteurs. En effet, pour combiner une mesure de profondeur fournie par un télémètre laser, avec une mesure fournie par un capteur à ultra-son, il nous faudra accorder plus de crédit à la valeur provenant



du télémètre. Ainsi, la profondeur résultant de la combinaison sera plus proche de la valeur télémétrique, que de celle produite par le capteur ultra-son.

Un second argument motive une prise en compte explicite de la notion de précision. La reconnaissance d'un objet connu dans la scène peut être sûre, si les indices de scène qui permettent d'en déduire l'existence sont précis. La reconnaissance de ce même objet peut être incertaine, si ces mêmes indices de scène sont estimés avec une forte imprécision. En observant une image photographique floue d'un groupe de personnes, on peut confondre les personnes qui se ressemblent physiquement. De même en robotique d'assemblage, une position incorrecte d'une pièce peut compromettre la bonne exécution d'une tâche d'insertion. Ces deux exemples montrent que l'exploitation des données sensorielles à un certain niveau, est conditionnée par la précision des données à un niveau inférieur.

Ces considérations nous ont amenés à représenter explicitement la précision des données numériques traitées par le système de perception.

La théorie des variables aléatoires multi-dimensionnelles offre un formalisme qui permet d'exprimer ce concept de précision. Certains travaux adoptent une approche comparable, en attribuant aux erreurs de mesure, une distribution statistique ([SC86], [AF87a],[Dur87], [RC87]).

Nous présentons tout d'abord la notion de variables aléatoires multi-dimensionnelles (v.a.m), puis nous expliquons comment cet outil permet de modéliser la précision de paramètres numériques. Enfin, nous montrons comment ce formalisme permet de résoudre trois problèmes importants, liés à la manipulation de la précision, dans un contexte de perception.

## 2.2 Les variables aléatoires multi-dimensionnelles (v.a.m)

Rappelons tout d'abord la définition d'une variable aléatoire à une dimension [Met72]:

**Une variable aléatoire  $X$  est une application d'un espace de probabilité  $(\Omega, \mathcal{A}, P)$  dans l'ensemble  $\mathbb{R}$  des réels, telle que pour tout couple  $(a, b)$  de réels, l'ensemble**

$\{a < X < b\}$  appartienne à la tribu  $\mathcal{A}$ .

De façon informelle, chaque élément de l'ensemble  $\Omega$  appelé éventualité, représente un aspect observable et indécomposable de ce que l'on peut observer d'un événement probable. Les éléments de la tribu  $\mathcal{A}$ , appelés événements, sont des sous-ensembles qui regroupent les éventualités, et représentent intuitivement ce qui peut se produire. L'ensemble  $\mathcal{A}$  vérifie certaines propriétés: par exemple, l'union de deux événements doit être un événement.

Une variable aléatoire est une fonction qui associe une valeur réelle à chaque éventualité, et permet donc de décrire les événements possibles en terme d'occurrences possibles de nombres, ou d'ensemble de nombres. La fonction de probabilité  $P$  permet de quantifier la "chance" qu'a chacun des événements de se produire, à l'aide d'un nombre réel compris entre 0 et 1. Lorsque  $P(e) = 1$ , l'événement  $e$  est "sûr". Lorsque  $P(e) = 0$ , l'événement  $e$  est "impossible".

Ainsi, pour tout segment  $[a, b]$  de  $\mathbb{R}$ ,  $P([a, b])$  quantifie la "chance" que l'événement ' $S = \text{la variable } X \text{ "tombe" dans } [a, b]$ ' se produise. Autrement dit,  $P([a, b])$  quantifie la "chance" que la variable  $X$  appartienne à  $[a, b]$ . Pour calculer cette valeur de probabilité  $P([a, b])$ , on définit une fonction de densité  $\zeta$  qui associe à chaque réel  $x$ , une valeur  $\zeta(x)$  qui permet d'évaluer la "chance" que possède la variable aléatoire  $X$  de se situer au voisinage de  $x$ . Ainsi,

$$P([x - \delta x, x + \delta x]) = 2\zeta(x)\delta x \quad (2.5)$$

Nous avons représenté sur la figure 2.1 une fonction de densité. La probabilité  $P([a, b])$  s'obtient en sommant toutes ces probabilités élémentaires  $P([x - \delta x, x + \delta x])$  sur l'intervalle  $[a, b]$ , ce qui mathématiquement, revient à calculer l'aire sous la courbe de  $\zeta$  entre  $a$  et  $b$ .

Parmi toutes les fonctions de probabilité, celles qui possèdent une densité normale (appelée aussi gaussienne) constituent une classe particulière intéressante. Une telle fonction  $P$  est définie de la façon suivante:

$$P([a, b]) = \int_a^b \zeta(x)dx \quad (2.6)$$

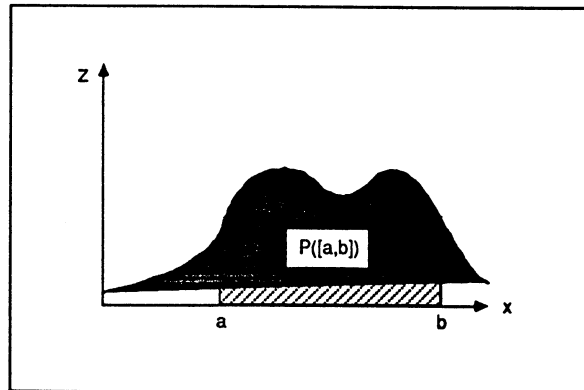


Figure 2.1: Calcul de  $P([a, b])$  à partir de la courbe de densité de probabilité

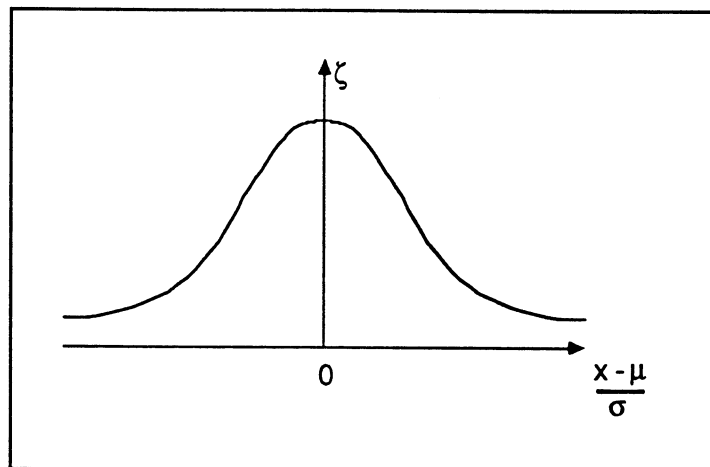


Figure 2.2: Densité gaussienne centrée réduite.

où:

$$\zeta(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.7)$$

est la densité de la fonction de probabilité  $P$ . Nous avons représenté figure 2.2 une densité gaussienne normale centrée réduite <sup>1</sup>.

En effet, les fonctions de probabilité de cette classe sont simples du point de vue mathématique, car leurs deux premiers moments suffisent à les caractériser entièrement.

**Le moment d'ordre 1:** Ou moyenne,  $\mu = E(X) = \int_{-\infty}^{+\infty} x\zeta(x)dx$ .

<sup>1</sup>“centrée” signifie que la moyenne  $\mu$  est nulle, “réduite” signifie que la variance  $\sigma$  est égale à 1.0 .

**Le moment d'ordre 2:** Ou variance,  $\sigma = E((X - \mu)^2) = \int_{-\infty}^{+\infty} (x - \mu)^2 \zeta(x) dx$ .

On peut noter la grande compacité de paramétrage des variables aléatoires gaussiennes. En effet, la donnée des deux paramètres  $(\mu, \sigma)$  détermine entièrement la variable aléatoire, par la donnée de sa densité de probabilité, définie à l'aide de la formule 2.7, et de sa fonction de probabilité, définie à l'aide de la formule 2.6.

Les variables aléatoires que nous avons définies ci-dessus, sont dites à une dimension, car elles prennent leurs valeurs dans  $\mathfrak{R}$ . La notion de variable aléatoire, s'étend naturellement au cas multi-dimensionnel.

Soient  $X_i$  ( $i \in \{1, \dots, n\}$ ),  $n$  variables aléatoires de dimension 1, on dit que le  $n$ -uplet  $(X_1, X_2, \dots, X_n)$  forme une variable aléatoire de dimension  $n$ . On parle alors de variables aléatoires multi-dimensionnelles, que l'on représente par la matrice colonne:

$$X = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}.$$

La notion de moyenne s'étend au  $n$ -uplet constitué des moyennes des  $n$  variables aléatoires, et se définit ainsi:

$$\mu = E\left(\begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}\right) = \begin{pmatrix} E(X_1) \\ \vdots \\ E(X_n) \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} \quad (2.8)$$

La notion de variance s'étend à la matrice de covariance. La matrice de covariance est définie ainsi:

$$\Lambda = E((X - \mu)(X - \mu)^{tr}) = \begin{bmatrix} E((X_1 - \mu_1)^2) & \cdots & E((X_1 - \mu_1)(X_n - \mu_n)) \\ \vdots & \ddots & \vdots \\ E((X_n - \mu_n)(X_1 - \mu_1)) & \cdots & E((X_n - \mu_n)^2) \end{bmatrix} \quad (2.9)$$

Cette matrice est aussi notée:

$$\Lambda = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1n}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{n1}^2 & \cdots & \sigma_n^2 \end{bmatrix} \quad (2.10)$$

C'est une matrice symétrique dont les coefficients  $\sigma_i^2$  sont les variances des variables aléatoires à une dimension  $X_i$ , et les coefficients  $\sigma_{ij}$  quantifient les corrélations existant entre les variables  $X_i$  et  $X_j$ . La densité de la fonction de probabilité (ou fonction de distribution) de  $P$  s'écrit alors:

$$\zeta(x) = \frac{1}{\sqrt{2\pi^n} \det(\Lambda)^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x - \mu)^{tr} \Lambda^{-1}(x - \mu)\right] \quad (2.11)$$

où

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \text{ et } \mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}$$

Comme dans le cas mono-dimensionnel, on définit la probabilité  $P("x \in \Delta")$  que  $x$  "tombe" dans un sous-ensemble fermé  $\Delta$  de  $\mathbb{R}^n$ , comme l'expression:

$$P("x \in \Delta") = \int_{\Delta} \zeta(x) dx \quad (2.12)$$

Dans le cas bi-dimensionnel ( $n = 2$ ), il est possible de représenter cette densité par une surface  $\zeta(x_1, x_2)$  (figure 2.3). Ce mode de représentation graphique n'est pas possible pour les dimensions supérieures. Par contre, il est possible de représenter graphiquement les ensembles de valeurs d'égale densité de probabilité.

Dans le cas bi-dimensionnel, ces ensembles sont des courbes, dont l'équation analytique se déduit de 2.11, et s'écrit:

$$\sigma_2^2(x_1 - \mu_1)^2 + \sigma_1^2(x_2 - \mu_2)^2 - 2\sigma_{12}^2(x_1 - \mu_1)(x_2 - \mu_2) = k^2(\sigma_1^2\sigma_2^2 - \sigma_{12}^2) \quad (2.13)$$

Ces ensembles constituent une famille d'ellipses, paramétrée par un réel  $k$ . Ces ellipses sont toutes centrées sur  $\mu = (\mu_1, \mu_2)$ . Nous avons représenté figure 2.4, l'ellipse correspondant à:

$$k = 1 ; \mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} ; \Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

Dans le cas tri-dimensionnel, l'ensemble d'iso-densité de probabilité forme un ellipsoïde d'équation:

$$\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \\ x_3 - \mu_3 \end{pmatrix}^{tr} \Lambda^{-1} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \\ x_3 - \mu_3 \end{pmatrix} = k^2 \quad (2.14)$$

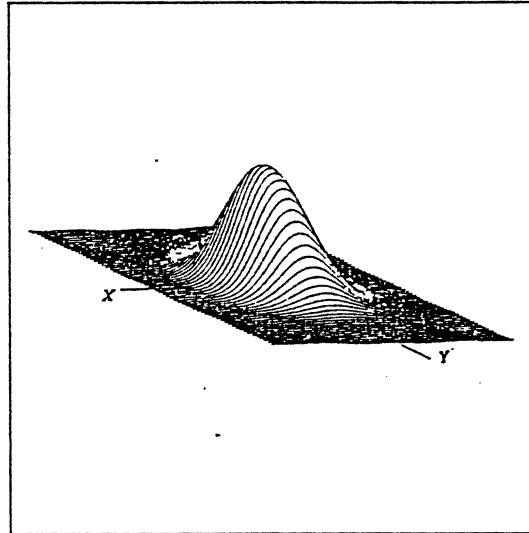


Figure 2.3: surface de densité gaussienne [J.L. Crowley, A representation for visual information, PhD thesis CMU, nov.1981]

où  $\Lambda$  est une matrice de covariance  $3 \times 3$ .

On peut alors interpréter la région intérieure à cet ensemble, comme l'ensemble des valeurs  $(x, y)$  possibles, avec une probabilité supérieure à une valeur  $f(k)$ . Une table permet d'exprimer la correspondance entre  $k$  et  $f(k)$ . Par la suite, nous fixerons la valeur de  $k$  à 1, afin de simplifier les formules (on a alors  $f(1) = 0.52$ ). Nous visualiserons donc la précision des variables mono- et multi-dimensionnelles, comme des régions dans lesquelles les variables ont une probabilité supérieure à 0.52 de se trouver.

## 2.3 Modélisation de la précision par les v.a.m

Définissons tout d'abord la notion de précision:

Soit un paramètre possédant une valeur exacte. Supposons qu'une méthode expérimentale permet de mesurer ce paramètre. La mesure est dite précise, si sa valeur diffère peu de la valeur exacte. La précision est maximale, lorsque la valeur mesurée est égale à la valeur exacte. Nous pouvons alors définir la précision d'une mesure, comme la différence entre la valeur mesurée et la valeur exacte. Cette différence

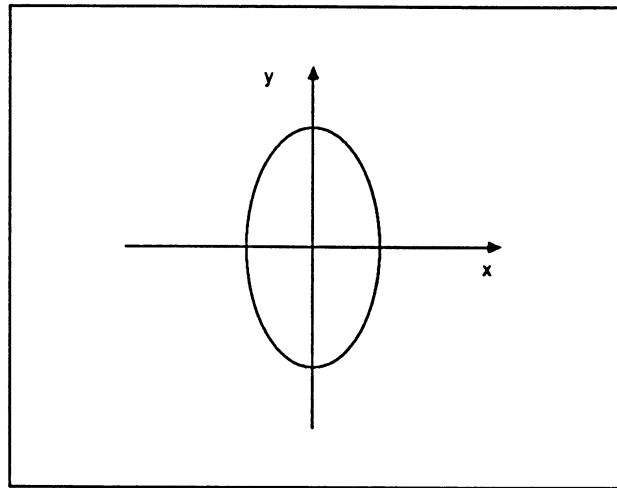


Figure 2.4: ellipse d'incertitude d'une v.a.m. bi-dimensionnelle

est aussi appelée erreur de mesure. En pratique, les valeurs exactes ne sont pas accessibles et leur mesure s'avère le seul moyen de les estimer. L'erreur de mesure n'étant pas, elle non plus, accessible, il est nécessaire de pouvoir la caractériser, afin de définir la précision de la mesure.

Il existe deux approches pour définir la précision: nous appellerons la première, approche "ensembliste". La deuxième utilise les v.a.m..

### 2.3.1 Approche ensembliste

Cette approche consiste à majorer l'erreur. La donnée de la mesure et du majorant de l'erreur, permet donc de définir un domaine, à l'intérieur duquel la valeur exacte est susceptible de se trouver.

Dans le cas mono-dimensionnel, ce domaine est constitué par un intervalle centré sur la valeur mesurée. Dans le cas multi-dimensionnel, on construit le plus petit volume englobant, parmi une classe de volumes facilement paramétrable. On utilise généralement des sphères ou des polygones. Cette approche est utilisée dans les travaux suivants: [Bro85], [Bro82], [Pug88].

L'inconvénient majeur de cette approche, réside dans la complexité des calculs et des formes de domaines lorsque la précision évolue au cours du traitement des mesures.

La solution qui consiste à approximer le domaine par un volume simple englobant à chaque étape, peut vite conduire à une précision “trop pessimiste” lorsque les erreurs s’accumulent

### 2.3.2 Approche par les sous-ensembles flous

Cette approche, qui est fondée sur la théorie des sous-ensembles flous, a été très peu étudiée. H. Farreny [FP85] en introduit les concepts et présentent les types de problème que la technique des sous-ensembles flous permet de résoudre.

Entre les trois approches, celle-ci impose moins de contraintes sur les propriétés de la précision. D’une part, l’ensemble des valeurs possibles ne possède pas forcément de frontières précises comme le suppose l’approche ensembliste. D’autre part, il n’est pas nécessaire de connaître la distribution des valeurs pour raisonner. De ce point de vue cette approche peut-être considérée comme plus générale que les deux autres. Malheureusement celle-ci se heurte à des difficultés techniques lorsque l’on veut l’appliquer à des variables de dimension supérieure à deux.

### 2.3.3 Approche par les v.a.m.

Cette troisième approche consiste à interpréter l’erreur comme la réalisation d’une variable aléatoire, et à caractériser cette variable aléatoire par sa distribution.

Lorsqu’on ne fait aucune hypothèse sur la fonction de distribution de la v.a.m., la première approche apparaît comme un cas particulier de v.a.m., qui posséderait une fonction de distribution uniforme, dont le support <sup>2</sup> est le domaine contenant les positions possibles de la valeur exacte.

Si on veut éviter les inconvénients liés aux premières approches que nous avons évoquées dans les paragraphes précédents, on doit émettre des hypothèses supplémentaires sur la fonction de distribution de la v.a.m.

La première hypothèse que nous faisons, admet l’existence d’un maximum de vraisemblance: nous supposons que le paramètre inconnu possède une valeur qui est plus probable que toutes les autres valeurs. En pratique, lorsque l’on réalise une mesure

---

<sup>2</sup>Le support d’une distribution est le sous-ensemble sur lequel elle n’est pas nulle.



d'un paramètre et que l'instrument de mesure n'est pas biaisé <sup>3</sup>, en l'absence de toute autre information, la valeur exacte la plus probable du paramètre est donnée par la mesure. La deuxième contrainte que nous nous imposons, est d'ordre technique. La fonction de distribution doit être facilement paramétrable, afin de rendre l'implémentation de la structure représentant l'imprécision aussi simple que possible.

Ces deux arguments nous ont conduit à n'utiliser que des variables aléatoires possédant une fonction de distribution gaussienne, pour modéliser la précision des paramètres numériques.

D'autres arguments ont conforté notre choix: une fonction de probabilité à densité normale représente un modèle satisfaisant de la dispersion de phénomènes physiques dans plusieurs situations réelles. Dans le domaine de classification des formes, la dispersion des formes observées autour de la forme type d'une classe, possède une distribution normale [Nil65]. Dans les domaines où l'on fait usage d'instruments de mesures, comme la vision par ordinateur, les erreurs de mesure peuvent être approximées par des variables aléatoires normales. En effet, ces erreurs résultent généralement d'une somme d'erreurs élémentaires indépendantes. Ces dernières s'accumulent au cours du traitement effectué par l'instrument de mesure, sur le signal acquis en entrée. Le "théorème central limite" démontre que la distribution de l'erreur finale tend vers une loi normale, même si les distributions des erreurs élémentaires ne sont pas normales.

Intéressons-nous maintenant aux différents traitements des variables aléatoires gaussiennes multi-dimensionnelles.

## 2.4 Manipulation de données imprécises

L'imprécision des données pose un certain nombre de problèmes, au système de perception. Les problèmes auxquels nous nous sommes intéressés sont les suivants:

**La comparaison de deux v.a.m.:** exemple: le système observe une surface plane à une distance de 1 mètre par l'intermédiaire de ses capteurs à ultra-son. Il

---

<sup>3</sup>Un instrument de mesure est biaisé si il produit une erreur systématique.

observe une surface située à 0.9 mètre, par l'intermédiaire de son capteur de proximité. Sachant que la précision des mesures obtenues par les capteurs ultra-son, est de 3cm, et que celle des mesures obtenues par le capteur de proximité est de 5%, le problème est de déterminer si ces mesures se rapportent au même plan physique.

**La combinaison de deux v.a.m.:** reprenons l'exemple précédent. Supposons que les deux mesures correspondaient bien au même plan physique. Le problème s'énonce ainsi: "Comment peut-on combiner les deux mesures pour évaluer plus précisément la profondeur du plan?"

**La propagation de la précision:** toujours avec le même exemple, supposons que le système possède en mémoire la position d'un objet posé sur le plan en question, ainsi que la précision de cette position. Problème: "Comment tenir compte de la réévaluation de la position du plan, pour mettre à jour la position de l'objet?"

Dans les paragraphes suivants, nous expliquons comment ces trois problèmes peuvent être résolus en pratique.

#### 2.4.1 Comparaison de deux variables aléatoires multi-dimensionnelles

Le problème de la comparaison de deux variables aléatoires se pose, lorsque l'on désire savoir si elles représentent toutes les deux le même paramètre réel.

Soit deux v.a.m.  $X = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}$  et  $Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$  de moyenne  $\mu_X$  et  $\mu_Y$ , et de

covariance  $\Lambda_X$  et  $\Lambda_Y$ . On définit une distance entre les deux v.a.m, quantifiant la probabilité qu'elles soient égales.

$$normdist(X, Y) = (\mu_X - \mu_Y)^{tr} (\Lambda_X + \Lambda_Y)^{-1} (\mu_X - \mu_Y) \quad (2.15)$$

**Justification:** Les variables  $X$  et  $Y$  étant indépendantes, la différence  $X - Y$  est une variable aléatoire de matrice de covariance  $\Lambda_X + \Lambda_Y$ , et de moyenne  $\mu_X - \mu_Y$ , qui suit une loi du  $\chi^2$  à

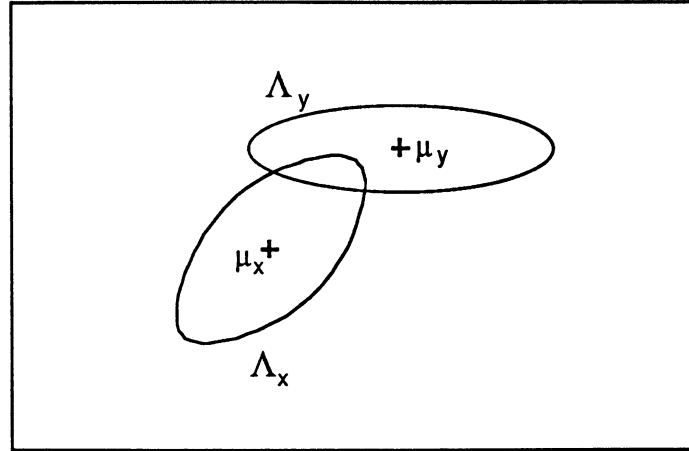


Figure 2.5: Comparaison de deux v.a.m. de dimension 2

deux degrés de liberté. La distribution gaussienne qui approxime le mieux la loi de distribution de  $X - Y$  s'écrit d'après la définition 2.11:

$$\zeta_{X-Y}(x) = \frac{1}{\sqrt{2\pi^n} \det(\Lambda_X + \Lambda_Y)^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x - \mu_X + \mu_Y)^{tr}(\Lambda_X + \Lambda_Y)^{-1}(x - \mu_X + \mu_Y)\right] \quad (2.16)$$

l'expression  $normdist(X, Y)$  définie dans la formule 2.15, est une fonction monotone de cette densité, puisqu'elle forme le terme en exposant dans l'expression de  $\zeta_{X-Y}(0)$ . Plus précisément,  $\zeta_{X-Y}(0)$ , qui est la densité de probabilité de l'événement: " $X - Y = 0$ " est d'autant plus grande, que  $normdist(X, Y)$  est petit. C'est pourquoi on peut utiliser  $normdist$  comme une mesure de distance entre  $X$  et  $Y$ . ( $normdist$  est aussi appelée "distance de Mahalanobis").

Si l'on revient aux propriétés des densités de probabilité des deux v.a.m., la relation:

$$normdist(X, Y) \leq 1 \quad (2.17)$$

permet d'affirmer que les deux v.a.m. peuvent être identiques avec une probabilité supérieure à 0.52. Dans le cas bi-dimensionnel, si on représente les ellipses de précision définies dans la section 2.2, le test 2.17 indique que les deux ellipses se recouvrent avec une probabilité supérieure à 0.52 (figure 2.5).

#### 2.4.2 Combinaison de variables aléatoires multi-dimensionnelles

Le problème abordé ici, est la suite logique de celui traité dans la section précédente.

Supposons que deux v.a.m.  $X$  et  $Y$  soient deux estimateurs indépendants d'un même paramètre réel. La combinaison de  $X$  et  $Y$  est une opération qui permet de construire une v.a.m., qui soit un "meilleur" estimateur du paramètre réel, que  $X$  et  $Y$ . Le critère d'optimalité utilisé est celui des moindres carrés. Celui-ci réalise une somme pondérée des carrés des erreurs, en affectant d'un poids important les mesures imprécises.

$$C(z) = (z - \mu_X)\Lambda_X^{-1}(z - \mu_X)^{tr} + (z - \mu_Y)\Lambda_Y^{-1}(z - \mu_Y)^{tr} \quad (2.18)$$

La recherche de l'estimateur optimum, consiste à trouver  $\mu_z$  qui minimise  $C(z)$ . Le critère étant quadratique, il nous suffit de différencier  $C(z)$  par rapport à  $z$ , et de chercher la valeur de  $z$  qui annule la différentielle.

$$\frac{\delta C}{\delta z} = (\Lambda_X^{-1} + \Lambda_Y^{-1})z - \Lambda_X^{-1}\mu_X - \Lambda_Y^{-1}\mu_Y$$

soit,

$$\mu_z = (\Lambda_X^{-1} + \Lambda_Y^{-1})^{-1}(\Lambda_X^{-1}\mu_X + \Lambda_Y^{-1}\mu_Y)$$

La définition 2.9 permet alors de calculer la matrice de covariance de  $Z$ :

$$\Lambda_Z = (\Lambda_X^{-1} + \Lambda_Y^{-1})^{-1}$$

### Combinaison de $n$ v.a.m

La technique de combinaison présentée ci-dessus dans le cas de deux v.a.m. , peut s'étendre au cas de trois v.a.m. ou plus.

Le problème s'énonce ainsi: à partir d'un ensemble  $\{X_i \mid i \in I\}$  de variables aléatoires multi-dimensionnelles qui estiment le même paramètre réel, on désire combiner ces v.a.m. afin de construire une nouvelle v.a.m. qui estime ce paramètre de façon optimale, au sens des moindres carrés.

Il existe deux approches à ce problème. La première s'inspire de la méthode utilisée dans le cas à deux v.a.m.. La deuxième est une méthode récursive appelée filtre de Kalman.

**Combinaison simple de  $n$  v.a.m.:** On cherche à minimiser le critère des moindres carrés:

$$C(x) = \sum_{i=1}^{i=n} (x - \mu_{X_i}) \Lambda_{X_i}^{-1} (x - \mu_{X_i})^{tr} \quad (2.19)$$

On montre facilement, en utilisant une démarche identique au cas  $n = 2$ , que la v.a.m. solution possède une moyenne  $\mu_x$ :

$$\mu_X = \left( \sum_{i=1}^{i=n} \Lambda_{X_i}^{-1} \right)^{-1} \left( \sum_{i=1}^{i=n} \Lambda_{X_i}^{-1} \mu_{X_i} \right) \quad (2.20)$$

et une matrice de covariance:

$$\Lambda_X = \left( \sum_{i=1}^{i=n} \Lambda_{X_i}^{-1} \right)^{-1} \quad (2.21)$$

**Combinaison de  $n$  v.a.m. à l'aide du filtre de Kalman:** le principe de cet algorithme consiste à calculer une combinaison optimale  $X^*$  des deux premières v.a.m.  $X_1$  et  $X_2$ , puis à combiner récursivement  $X^*$  avec les v.a.m.  $X_i$  suivantes.

La première étape de la combinaison de deux v.a.m.  $X_1$  et  $X_2$ , consiste à calculer le *facteur de gain*  $K$  du filtre:

Soit  $\Lambda_{X_i}$  la matrice de covariance de  $X_i$ , alors

$$K = \Lambda_{X_1} (\Lambda_{X_1} + \Lambda_{X_2})^{-1} \quad (2.22)$$

La matrice de covariance de la v.a.m optimale  $X^*$  s'écrit alors:

$$\Lambda_{X^*} = \Lambda_{X_1} - K \Lambda_{X_2}^{tr} \quad (2.23)$$

La valeur moyenne de la v.a.m. optimale  $X^*$  est donnée par la formule:

$$\mu_{X^*} = \mu_{X_1} + K(\mu_{X_2} - \mu_{X_1}) \quad (2.24)$$

Si on appelle *combine* l'application qui aux deux variables  $X_1$  et  $X_2$  associe la variable aléatoire  $X^*$ , l'algorithme global du filtre de Kalman, qui permet de calculer la v.a.m optimale  $X^*(E)$  par combinaison d'un ensemble  $E = \{X_i \mid i \in I\}$ , s'écrit donc:

$$\begin{aligned} \text{si } E = \{X_1, X_2\} \text{ alors } X^*(E) &= \text{combine}(X_1, X_2) \\ \text{sinon } X^*(E) &= \text{combine}(X_n, X^*(E - \{X_n\})) \end{aligned}$$

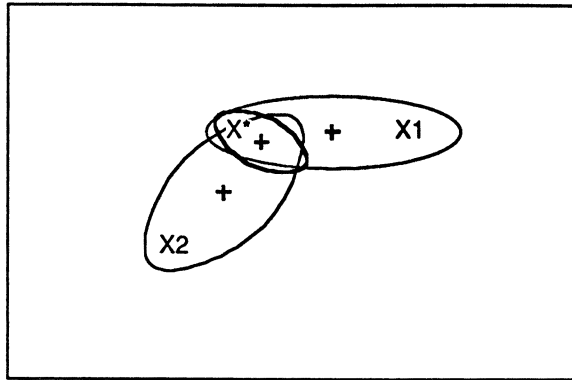


Figure 2.6: Combinaison de deux v.a.m. de dimension 2

Par rapport à la méthode simple de combinaison, la méthode du filtre de Kalman présente l'avantage de fournir à chaque étape d'itération  $k$  ( $1 < k < n$ ), un estimateur  $(X^*, \Lambda_{X^*})$  qui tient compte des  $k$  premières v.a.m.

Nous avons représenté graphiquement dans la figure 2.6, le résultat d'une combinaison de deux variables aléatoires de dimension 2. On constate bien que l'ellipse d'incertitude de la v.a.m. obtenue, approxime l'intersection des ellipses d'incertitude des deux v.a.m..

Jusqu'ici, nous avons combiné des variables aléatoires, qui estiment la même entité physique, pour construire une nouvelle v.a.m. qui constitue un meilleur estimateur. Dans certains cas, le problème est différent. On dispose de deux v.a.m. indépendantes, dont on sait que les entités qu'elles estiment sont liées par une contrainte d'ordre physique. On veut réestimer ces entités en tenant compte de cette contrainte.

#### Exemple:

Dans un espace 2D, on considère un point  $P$  et une droite  $D$ . Le point est estimé avec une certaine précision indiquée par une ellipse sur la figure 2.7 a. La droite est estimée indépendamment, avec une précision représentée par une bande. On apprend, de source indépendante, que le point  $P$  est situé sur la droite  $D$ . On veut utiliser cette nouvelle information, pour réestimer le point  $P$  et la droite  $D$  (voir figure 2.7 b)

Dans le paragraphe suivant, nous présentons la technique du filtre de Kalman

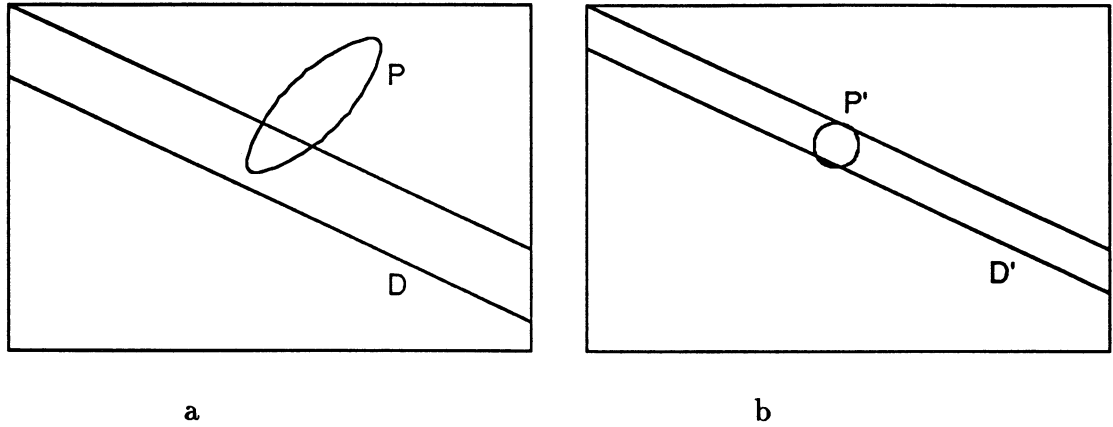


Figure 2.7: Réestimation de deux v.a.m. de natures différentes

étendu, qui permet de résoudre ce problème, lorsque la contrainte peut s'exprimer à l'aide d'une équation analytique de la forme  $f(P, D) = 0$ .

### Combinaison de deux v.a.m. de nature différentes

La technique utilisée est détaillée dans [Aya88].

Le problème est le suivant: on dispose de deux variables aléatoires  $X$  et  $Y$ , définie chacune par une moyenne  $\mu_X$ ,  $\mu_Y$ , et une matrice de covariance  $\Lambda_X$ ,  $\Lambda_Y$ . D'une source d'information indépendante de celle qui a permis d'obtenir ces valeurs, on établit une contrainte s'exprimant sous forme d'une équation linéaire de la forme:

$$AX = BY + C \quad (2.25)$$

où les matrices  $A$  et  $B$  permettent de transformer les variables  $X$  et  $Y$  dans le même espace, et de les comparer. Nous voulons réestimer les moyennes  $\mu_X^*$ ,  $\mu_Y^*$  et les matrices de covariances  $\Lambda_X^*$ ,  $\Lambda_Y^*$  des variables  $X$  et  $Y$ , en tenant compte de l'information contenue dans la contrainte 2.25.

La technique du filtre de Kalman étendu, permet de résoudre ce problème. Nous présentons les formules du filtre. Pour plus de détails, on pourra se reporter à l'ouvrage [LKG78].

La nouvelle valeur moyenne de  $X$  s'obtient par la formule:

$$\mu_X^* = \mu_X + K(B\mu_Y + C - A\mu_X) \quad (2.26)$$

où  $K$  est la matrice de "gain", calculée par la formule:

$$K = \Lambda_X A^{tr} (B \Lambda_Y B^{tr} + A \Lambda_X A^{tr})^{-1} \quad (2.27)$$

enfin, la matrice de covariance de  $X^*$  s'obtient par:

$$\Lambda_X^* = (I_n - KA) \Lambda_X \quad (2.28)$$

où  $I_n$  dénote la matrice identité de même dimension que  $\Lambda_X$ .

Pour calculer  $\mu_Y^*$  et  $\Lambda_Y^*$ , il suffit simplement d'inverser les rôles joués par  $(X, A)$  et  $(Y, B)$  dans les formules précédentes.

**Lorsque la contrainte n'est pas linéaire** Dans le cas où la contrainte entre  $X$  et  $Y$ , s'exprime à l'aide d'une équation non linéaire de la forme:

$$f(X, Y) = 0 \quad (2.29)$$

Si la fonction  $f$  est suffisamment régulière au voisinage de la solution, on peut faire une approximation linéaire de l'équation 2.29 au voisinage de  $\mu_X$  et  $\mu_Y$ , en effectuant un développement de Taylor:

$$f(\mu_X, \mu_Y) + \frac{\delta f}{\delta X}(X - \mu_X) + \frac{\delta f}{\delta Y}(Y - \mu_Y) = 0 \quad (2.30)$$

On peut alors appliquer la technique précédente en posant:

$$\begin{aligned} A &= \frac{\delta f}{\delta X} \\ B &= -\frac{\delta f}{\delta Y} \\ C &= \frac{\delta f}{\delta X} \mu_X + \frac{\delta f}{\delta Y} \mu_Y - f((\mu_X, \mu_Y)) \end{aligned} \quad (2.31)$$

### 2.4.3 Transformation de variables aléatoires multi-dimensionnelles

De même qu'on définit des transformations  $\left\{ \begin{array}{ccc} \mathcal{R} & \xrightarrow{f} & \mathcal{R} \\ x & \mapsto & y = f(x) \end{array} \right.$  entre des réels, ou

des transformations  $\left\{ \begin{array}{ccc} \mathcal{R}^n & \xrightarrow{f} & \mathcal{R}^m \\ \left( \begin{array}{c} x_1 \\ \vdots \\ x_n \end{array} \right) & \mapsto & \left( \begin{array}{c} y_1 \\ \vdots \\ y_m \end{array} \right) \end{array} \right.$  entre des n-uplets de réels, on définit

des transformations entre les v.a.m.

$$X \mapsto Y = f(X)$$



Généralement, ces transformations sont linéaires ( $Y = AX$ ). Lorsqu'elles ne le sont pas, il faut se ramener à la transformation linéaire tangente, dont la matrice  $A$  est obtenue en calculant le jacobien de la transformation  $f$ .

On a alors les relations suivantes entre les moyennes et les matrices de covariance des v.a.m.  $X$  et  $Y$  :

$$\mu_Y = A\mu_X \text{ (ou } \mu_Y = f(\mu_X)) \quad (2.32)$$

$$\Lambda_Y = A\Lambda_X A^{tr} \text{ (ou } \Lambda_Y = \frac{\delta f}{\delta X} \Lambda_X \frac{\delta f}{\delta X}^{tr} \text{ )} \quad (2.33)$$

Démonstration:

Par définition:

$$\Lambda_Y = E((Y - \mu_Y)(Y - \mu_Y)^{tr})$$

soit,

$$\begin{aligned} \Lambda_Y &= E((AX - A\mu_X)(AX - A\mu_X)^{tr}) \\ &= E(A(X - \mu_X)(X - \mu_X)^{tr} A^{tr}) \\ &= AE((X - \mu_X)(X - \mu_X)^{tr}) A^{tr} \end{aligned}$$

finalement,

$$\Lambda_Y = A\Lambda_X A^{tr}$$

## 2.5 Résumé

Nous avons présenté le modèle probabiliste des variables aléatoires multi-dimensionnelles (v.a.m.).

Nous avons montré comment un vecteur de mesure imprécis pouvait être représenté par une variable aléatoire multi-dimensionnelle. Sous cette optique, la moyenne de la v.a.m. représente la valeur de la mesure. La matrice de covariance de la v.a.m. caractérise la dispersion de l'erreur de mesure. La densité de la v.a.m. fournit donc une carte indiquant les lieux, dans lesquels la valeur exacte possède une forte probabilité de se situer.

L'utilisation des v.a.m. pour modéliser la précision, nous permet de trouver des solutions pratiques, face à trois problèmes typiques de l'intégration multisensorielle:

- Nous avons défini la distance “*normdist*” entre deux v.a.m., qui permet de décider si les mesures imprécises qu’elles modélisent, peuvent constituer deux mesures différentes d’un même paramètre réel.
- Nous avons introduit des techniques permettant de combiner plusieurs v.a.m., pour construire un estimateur optimal, à partir de plusieurs mesures imprécises que ces v.a.m. modélisent.
- Enfin, nous avons décrit une méthode permettant de relier la précision d’un paramètre inconnu, à la précision d’une mesure qui lui est reliée par une formule analytique.



## Chapitre 3

# modélisation géométrique

### 3.1 Introduction

Pour définir ses actions sur l'environnement extérieur, un robot doit utiliser sa connaissance de la géométrie des éléments qui composent cet environnement. Pour se déplacer et éviter les collisions, il doit tenir compte de la position des obstacles, et du volume que ceux-ci occupent dans l'espace. Pour saisir un objet, il doit rechercher les surfaces de l'objet qui peuvent être mises en contact avec les mors de l'outil de préhension, et ensuite définir une trajectoire d'approche, qui n'interfère ni avec les autres parties de l'objet, ni avec sa propre géométrie.

Ces exemples montrent que les informations géométriques jouent un rôle important dans la façon dont le robot planifie ses actions. Par informations géométriques, nous entendons la connaissance quantitative des dimensions, des orientations et des positions des éléments constituant l'environnement. Une représentation de ces informations est appelée **modélisation géométrique**.

Il n'existe pas de modélisation géométrique universelle, qui soit adaptée à toutes les applications. Généralement, le choix du mode de représentation des informations géométriques dépend essentiellement de l'utilisation que l'on veut en faire. Nous décrivons les différents modes de représentation utilisés, puis nous définirons et justifierons notre choix.

### 3.1.1 Ce qui existe

On peut classer les modes de représentation en trois types (voir [Per86]): les représentations par échantillonnage de l'espace, les représentations par volumes, et les représentations surfaciques.

#### Représentation par échantillonnage

Dans ce type de représentation, on peut énumérer l'espace occupé. Pour cela, on procède au découpage de l'environnement en cubes de taille fixe. Chaque cube est étiqueté suivant qu'il est occupé ou non [BB82].

On peut aussi décomposer récursivement les objets en cellules. L'arrêt dans la récursion se produit lorsque la cellule est entièrement contenue dans la matière, ou entièrement extérieure l'objet, ou enfin lorsque la cellule est de taille minimale. La structure de la représentation obtenue est arborescente (voir figure 3.1).

#### Représentation par volumes

Dans ce type de modélisation, les objets sont obtenus par combinaison de volumes élémentaires simples. Par exemple, des cubes, des cônes, des cylindres ([RV79]), ou même des volumes un peu plus complexes, comme les cylindres ou les cônes généralisés, obtenus par balayage d'une section plane sur une trajectoire paramétrée ([BGB79], [AB85]). Ces volumes sont combinés à l'aide des opérateurs booléens (union, intersection, différence), pour former des volumes plus complexes. Ces derniers peuvent être eux-mêmes recombinaisonnés. Ces opérations de combinaisons sont répétées jusqu'à l'obtention de l'objet. Ce type de représentation est essentiellement utilisé en C.A.O figure 3.2.

Dans le système ACRONYM [BGB79], la description des objets peut être générique.

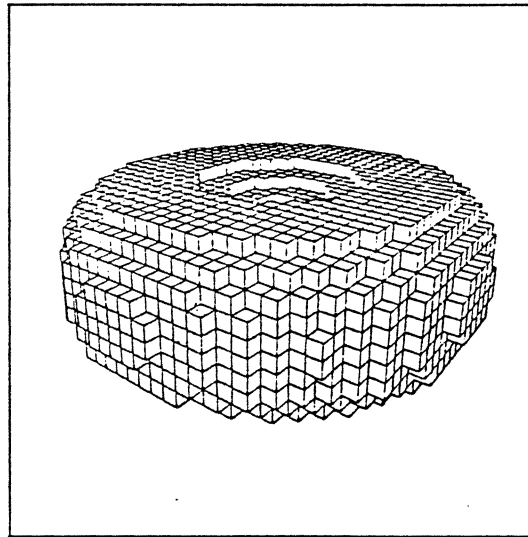


Figure 3.1: Représentation par échantillonnage de l'espace [D.H.Ballard and C.M.Brown, Computer vision, Prentice Hall 1982]

En effet, lorsque les dimensions des volumes élémentaires ne sont pas spécifiées, la structure ne décrit pas un objet unique, mais une classe d'objets.

### Représentation par surfaces

Dans ce type de modélisation, les objets sont représentés par leur enveloppe extérieure. Cette enveloppe est décrite en termes de surfaces, d'arêtes, de points (voir figure 3.3). C'est généralement ce type de représentation qui est utilisé dans les travaux sur la vision par ordinateur.

On pourra trouver une étude détaillée des représentations de formes 3D en vision par ordinateur dans [Fau82].

#### 3.1.2 Nos contraintes et nos choix

Différentes considérations nous ont guidé dans le choix d'un mode de représentation des informations géométriques. La première concerne la puissance de représentation. Les suivantes concernent la structuration de l'information qui est sous-jacente à cette représentation.

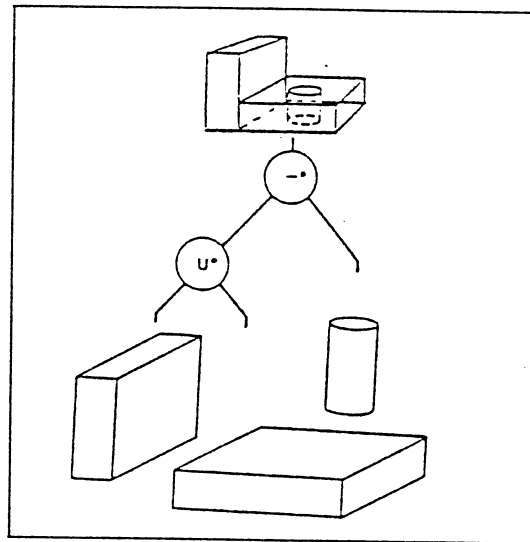


Figure 3.2: Représentation par volumes [A.G. Requicha and H.B. Voelckert, Solid Modelling: a historical summary and contemporary assessment, IEEE Computer Graphics and its applications, 1982]

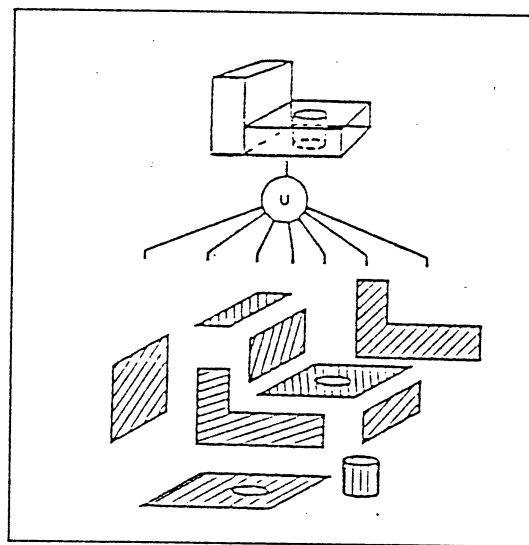


Figure 3.3: Représentation par surfaces [A.G. Requicha and H.B. Voelckert, Solid Modelling: a historical summary and contemporary assessment, IEEE Computer Graphics and its applications, 1982]

1. La souplesse: la représentation doit être apte à décrire les informations fournies par la plupart des capteurs existants.
2. L'extensibilité: le mode de représentation doit pouvoir s'enrichir de types d'information nouveaux, dont nous envisageons l'intégration dans le futur (couleurs, textures, formes surfaciques complexes,...).
3. La disponibilité: les données doivent être accessibles rapidement et efficacement, notamment lorsque la masse d'information est importante. Cette contrainte suggère un type d'accès à l'information par "association".

Ces contraintes nous ont amenés à nous orienter vers les solutions suivantes, que l'on peut classer dans le groupe des représentations par surfaces:

1. les informations géométriques s'expriment à l'aide de primitives paramétriques. Une primitive géométrique permet de décrire localement la géométrie de l'environnement, sous forme d'une entité géométrique simple qui l'approxime. Nous définirons ces primitives dans le détail au paragraphe 3.2. Cette représentation permet une certaine extensibilité, car pour décrire un nouveau type de surface, il suffira d'introduire une nouvelle primitive.
2. Les relations géométriques sont également représentées. La prise en compte des relations géométriques permet d'une part d'accéder facilement à l'information locale contenue dans les primitives géométriques. D'autre part, les relations géométriques procurent une source importante d'informations.

Nous pouvons remarquer que la notion d'objet, qui existe dans la plupart des modes de représentation, et qui permet de regrouper les primitives géométriques en entités distinctes, n'apparaît pas dans notre description. Pourtant, lors d'une phase de manipulation, la définition explicite des objets est indispensable au robot pour lui permettre de trouver une prise correcte, et de tenir compte du déplacement de tous les éléments géométriques appartenant à l'objet manipulé, dans la recherche de sa trajectoire. Les choix que nous avons adopté jusqu'ici, nous laissent entrevoir deux amorces de solution à ce problème.

1. Ce niveau objet est rajouté explicitement à la description interne de l'environnement. On confie alors au système de perception la tâche d'extraire les objets



connus de l'ensemble des primitives. Cette solution nous paraît inintéressante à deux points de vue. D'une part, la charge de travail du système de perception s'alourdit de façon conséquente. De plus se présente le risque de voir le système reconnaître plus d'objets qu'il n'est réellement nécessaire.

2. Ce niveau objet est construit dans une structure séparée de la description interne. Le principe est de reconnaître les objets dont on a besoin, et quand on en a besoin. Cette solution permet d'affecter les tâches de perception et de reconnaissance à deux modules distincts, pouvant éventuellement fonctionner en parallèle.

Au stade actuel de notre réflexion, nous n'avons pas pris de décision à ce sujet, mais sommes pleinement conscients de la nécessité d'explicitier la notion d'objet pour franchir le fossé entre perception et manipulation.

Notons néanmoins, que le choix d'un mode de représentation ne compromet pas de manière irrémédiable la possibilité de réaliser des tâches pour lesquelles celle-ci n'est pas bien adaptée. En effet, il est parfois possible de passer d'un mode de représentation à l'autre ([RV82]). Des travaux dans ce sens sont menés au LIFIA.

## 3.2 Primitives géométriques

Actuellement, nous envisageons d'utiliser trois types de primitives géométriques: les sommets 3D, les segments 3D, les éléments de surface.

### 3.2.1 Les sommets 3D

Un sommet  $P$  est un point de l'espace vers lequel concourent plusieurs segments 3D. Les sommets 3D nous permettent de décrire les coins physiques des objets localement polyédriques. Il est caractérisé par:

**Des coordonnées:**  $(x, y, z)$  qui fournissent une position estimée du point par rapport au repère absolu de la scène.

**Une matrice de covariance:**  $\Lambda_P$  qui définit la précision de la position estimée  $(x, y, z)$ . C'est une matrice de dimension  $3 \times 3$ .

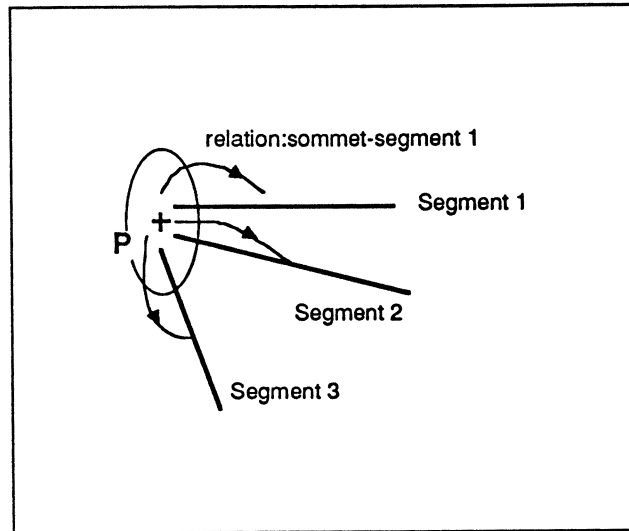


Figure 3.4: Sommet 3D

Une liste de relations: R-list qui enregistre les relations auxquelles  $P$  participe.

Voir figure 3.4

### 3.2.2 Les segments 3D

Un segment 3D  $S$  est une primitive géométrique qui nous permet de décrire les arêtes des objets localement polyédriques. Un segment 3D est caractérisé par:

**Les coordonnées  $(x, y, z)$  de son point milieu  $P$**

**La matrice de covariance  $\Lambda_P$  qui définit la précision de  $(x, y, z)$ .**

**Les coordonnées  $(x_N, y_N, z_N)$  de son vecteur directeur . Le vecteur directeur  $N$  est normé.**

**La matrice de covariance  $\Lambda_N$  qui définit la précision de  $(x_N, y_N, z_N)$ .**

**Une liste R-liste** contenant les relations auxquelles le segment  $S$  participe.

Ces attributs définissent entièrement le segment 3D, mais nous utilisons aussi les attributs redondants suivants, afin de rendre plus efficace l'exécution de certaines procédures:

**Les coordonnées des extrémités  $(x_1, y_1, z_1)$  et  $(x_2, y_2, z_2)$**

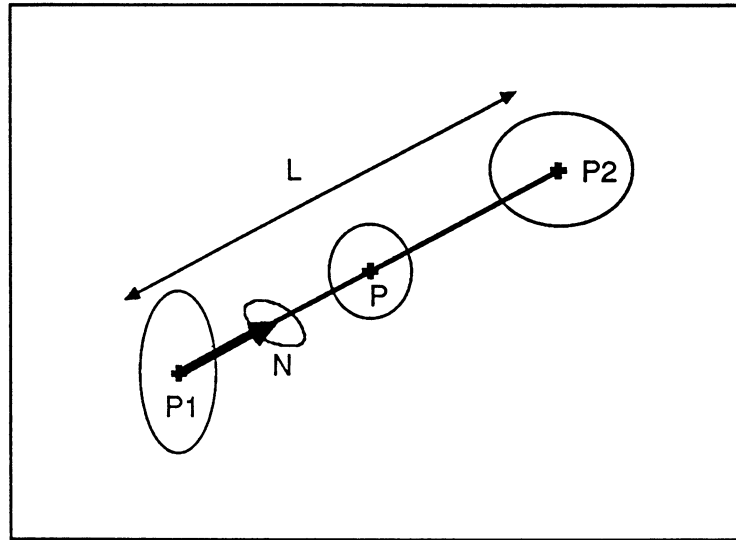


Figure 3.5: Segment 3D

**Les matrices de covariance**  $\Lambda_{P_1}$  et  $\Lambda_{P_2}$  associées aux vecteurs  $(x_1, y_1, z_1)$  et  $(x_2, y_2, z_2)$ .

**La longueur L** du segment

Voir figure 3.5

### 3.2.3 Les éléments de surface

Un élément de surface  $ES$  est une primitive géométrique qui permet de décrire localement les portions de surface plane des objets. Cette primitive est définie par les attributs suivants:

**Les coordonnées**  $(x_P, y_P, z_P)$  d'un point  $P$  de la surface

**La matrice de covariance**  $\Lambda_P$ , qui définit la précision attachée aux coordonnées  $(x_P, y_P, z_P)$  du point  $P$ .

**Les coordonnées**  $(x_N, y_N, z_N)$  du vecteur normal à la surface. Ce vecteur est normalisé et orienté vers l'extérieur de l'objet.

**La matrice de covariance**  $\Lambda_N$ , qui définit la précision du vecteur normal  $N$ .

**Une liste R-liste** qui contient les relations auxquelles  $ES$  participe.

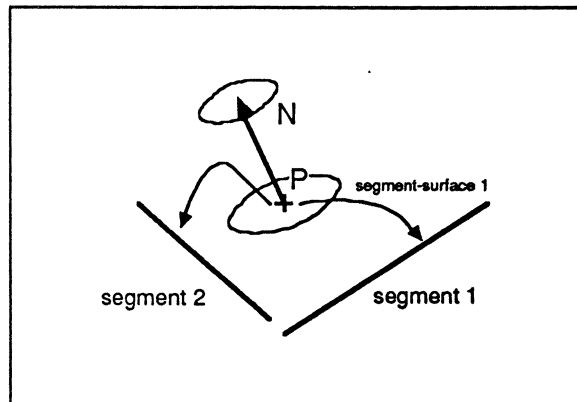


Figure 3.6: Élément de surface

Voir figure 3.6.

Nous décrivons dans le paragraphe suivant la structure des relations géométriques qui permettent de relier les primitives géométriques entre elles.

### 3.3 Relations géométriques

Chaque relation géométrique forme un objet dont un des attributs est le type de relation géométrique. Les autres attributs pointent vers les primitives que la relation unit. Nous aurions pu représenter les relations géométriques de façon implicite, en définissant au niveau de la primitive, des attributs pointant directement vers les autres primitives auxquelles elle est reliée géométriquement.

Deux arguments nous ont déterminé à représenter explicitement les relations géométriques.

- Une relation géométrique constitue une information géométrique. En tant que telle, cette information peut être remise en question, sans que les primitives qui partagent cette relation soient remises en cause.
- Cette représentation explicite permet une plus grande extensibilité de la modélisation. Si nous voulons introduire une nouvelle primitive géométrique, il nous suffit de définir une nouvelle structure et ses attributs, puis les relations qui font intervenir cette primitive comme une nouvelle structure de données.

### 3.3.1 Les relations d'appartenance

#### Segment-Segment

la relation est définie par les attributs:

**type:** appartenance-segment-segment.

$S_1$ : est un pointeur qui désigne un segment 3D.

$S_2$ : est un pointeur qui désigne un autre segment 3D.

#### Segment-Surface

La relation est définie par les attributs:

**type:** appartenance-segment-surface.

$S$ : est un pointeur qui désigne un segment 3D.

$ES$ : est un pointeur qui désigne un élément de surface 3D.

#### Sommet-Surface

La relation est définie par les attributs:

**type:** appartenance-sommet-surface.

$P$ : est un pointeur qui désigne un sommet 3D.

$S$ : est un pointeur qui désigne un segment 3D.

#### Sommet-Segment

La relation est définie par les attributs:

**type:** appartenance-sommet-segment.

$P$ : est un pointeur qui désigne un sommet 3D.

$S$ : est un pointeur qui désigne un segment 3D.

### 3.3.2 Les relations de parallélisme

#### Segment-Segment

La relation est définie par les attributs:

**type:** parallèle-segment-segment

$S_1$ : est un pointeur qui désigne un segment 3D.

$S_2$ : est un pointeur qui désigne un segment 3D.

#### Segment-Surface

La relation est définie par les attributs:

**type:** parallèle-segment-surface

$S$ : est un pointeur qui désigne un segment 3D.

$ES$ : est un pointeur qui désigne un élément de surface 3D.

#### Surface-Surface

La relation est définie par les attributs:

**type:** parallèle-surface-surface

$S_1$ : est un pointeur qui désigne un élément de surface 3D.

$S_2$ : est un pointeur qui désigne un second élément de surface 3D.

La liste de relations géométriques que nous avons définies n'est pas exhaustive. Nous nous sommes limités aux relations d'appartenance et de parallélisme, car elles sont suffisamment générales et indépendantes du contexte, pour être représentées explicitement et "statiquement" dans le modèle interne de l'environnement. Les autres relations géométriques qui pourraient être nécessaires dans un environnement spécifique, ou pour une tâche particulière, peuvent être établies "dynamiquement" et temporairement pendant l'exécution même de la tâche. Par exemple, lorsque l'agent veut reconnaître un objet polyédrique, la relation de perpendicularité des arêtes ou des faces est une contrainte qui peut être très discriminante pour sélectionner parmi

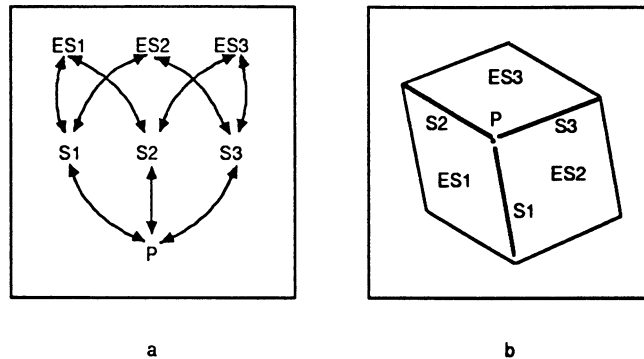


Figure 3.7: exemple de graphe de relations géométriques

les éléments de la représentation interne, ceux qui peuvent appartenir à l'objet cherché. Ces relations de perpendicularité sont alors enregistrées dans des structures temporaires, puis détruites une fois la tâche de reconnaissance achevée.

Les différents éléments que nous avons défini jusqu'ici, induisent une structuration de l'information géométrique, que nous décrivons dans le paragraphe suivant.

### 3.4 Structure globale de l'information géométrique

Les primitives géométriques sont organisées en listes, qui regroupent les primitives de même type. Par dessus cette organisation "par types", se plaque une organisation en "graphe". En effet, l'ensemble des primitives géométriques et des relations géométriques forme un graphe qui saisit l'information géométrique d'un point de vue global. Nous avons représenté figure 3.7 a, le graphe correspondant à la configuration de la figure 3.7 b.

Cette structure organise les données de façon efficace, en permettant une exploration intelligente de l'information. Si par exemple, on désire rechercher les arêtes coplanaires d'une arête donnée  $S$ , il suffit de chercher dans sa liste de relations  $S \rightarrow R - \text{liste}$ , les relations du type "parallèle-segment-surface". Puis pour chacune de ces relations  $R_1$  de ce type, on extrait l'élément de surface  $R_1 \rightarrow ES$ . Les arêtes que l'on cherche sont contenues dans  $R_1 \rightarrow ES$ . On cherche donc dans la liste  $R_1 \rightarrow ES \rightarrow R - \text{liste}$ , les relations  $R_2$  du type "parallèle-segment-surface". Pour les relations  $R_2$  de ce type trouvées, le segment  $R_2 \rightarrow S$  est coplanaire à  $S$ .

### 3.5 Résumé

Nous avons choisi un mode de représentation géométrique, basé sur un ensemble de primitives géométriques simples et de relations géométriques permettant de relier les primitives entre elles. Ces primitives comprennent:

**les sommets 3D** qui permettent de décrire les coins des objets localement polyédriques.

**les segments 3D** qui permettent de décrire les arêtes des objets.

**les éléments de surface** qui permettent de décrire les parties localement planes des objets.

Les relations géométriques comprennent:

**les relations d'appartenance** qui permettent de traduire l'appartenance de sommets et de segments à un autre segment, et l'appartenance de sommets, de segments et de surfaces à une autre surface.

**les relations de parallélisme** qui permettent de traduire le parallélisme entre contours et le parallélisme entre surfaces.

Ces primitives géométriques sont caractérisées par des attributs, qui définissent quantitativement leur position et leur orientation dans l'espace, ainsi que des attributs qui définissent la précision de ces informations quantitatives. Les primitives géométriques définissent localement la géométrie de l'environnement. Les relations géométriques sont des objets qui possèdent des attributs permettant de désigner les primitives qu'elles relient.

Ces primitives géométriques sont regroupées par types, dans des listes. Par dessus cette structure, le graphe des relations géométriques permet d'organiser les primitives géométriques en un réseau associatif, et offre une représentation globale de l'environnement.

L'accès à l'information et son exploration peuvent s'envisager de différentes manières: soit par type de primitives, en parcourant la liste des primitives, soit en utilisant la configuration du voisinage de la primitive dans le graphe des relations géométriques,



et en guidant l'exploration par les relations.

Cette structure permet d'accéder efficacement à l'information pertinente, en utilisant des contraintes géométriques.

## Chapitre 4

# Représentation et gestion de l'incertitude

### 4.1 Introduction

Nous avons présenté dans un chapitre précédent, la notion d'incertitude géométrique. L'incertitude géométrique, que nous préférons nommer “imprécision”, est le phénomène qui permet d'expliquer l'écart existant entre la valeur réelle d'un paramètre et sa valeur estimée.

Dans ce chapitre, nous étudions une autre notion de l'incertitude, qui est associée à l'existence même des données. Pour l'introduire, nous soulignons la nécessité pour un système de perception de prendre en compte des **informations incertaines** ou **contradictaires**. Nous introduisons le concept d'**hypothèse**, comme un moyen de représenter cette notion d'incertitude. Nous précisons la structure d'une hypothèse, ainsi que les opérations primitives qui permettent la **gestion d'un ensemble**

d'hypothèses.

## 4.2 La perception face à l'incertitude et la contradiction

Rappelons que la fonction de la perception est de fournir à l'agent qui en est doté, une description interne de son environnement. Cette description est construite à partir de mesures brutes que lui fournissent des capteurs. On appelle interprétations les éléments de la description interne.

Plusieurs facteurs concourent à rendre ambiguës, ou à fragiliser les interprétations générées par la perception :

- Les capteurs fournissent des données incomplètes: la formation d'une image visuelle par exemple, "écrase" toute information de profondeur par projection perspective.
- les mesures peuvent être partiellement ou complètement erronées.
- Les données peuvent être contradictoires: deux capteurs différents peuvent fournir des mesures en complet désaccord si l'un des capteurs ne fonctionne pas correctement.
- Les mesures relèvent de multiples phénomènes physiques qu'il est généralement impossible de dissocier: les lignes de contraste dans une image peuvent être dues à des arêtes réelles d'objets (courbes de discontinuité de la surface), ou aux contours d'une zone d'ombre, ou encore aux bords d'un reflet... Cette multitude d'explications conduit à une multitude d'hypothèses, lorsque l'on veut "interpréter" l'image en termes d'indices de scène. La plupart du temps, une seule de ces interprétations est vraie. Une ligne de contraste, par exemple, ne peut être expliquée à la fois, par une arête, et par le contour d'une zone d'ombre. Les interprétations peuvent alors être contradictoires.

Ces considérations nous ont amenés à définir un cadre formel pour représenter des informations incertaines et contradictoires, et pour gérer ces informations du point de vue de leur cohérence globale, et de leur durée de vie. Ce sont ces points que nous détaillons dans les paragraphes suivants:

Certains points développés dans ce chapitre, ont été introduits dans l'article [dK86] (exploration simultanée des alternatives), et l'article [Doy77] (aspects de la non-monotonie).

### 4.3 L'hypothèse: une représentation de l'incertitude

Il nous faut expliquer la façon dont la notion d'incertitude s'intègre à la modélisation déjà introduite. Il nous faut également décrire les concepts que nous voulons représenter dans ce formalisme. Ces concepts sont les suivants:

- Une mesure associée à chaque élément d'information, afin d'évaluer le degré de confiance qu'on peut lui accorder.
- L'incompatibilité entre différents éléments d'information.
- La dépendance entre éléments d'information. Cette notion permet de prendre en compte, le fait que la perception construit des interprétations, à partir d'interprétations antérieures.

#### 4.3.1 Définition d'une hypothèse

La représentation des hypothèses que nous utilisons s'adapte parfaitement à notre modélisation géométrique de l'environnement. En effet, elle pourrait convenir à tous les domaines dans lesquels l'information est discrète. Plus précisément, une hypothèse est une structure de données, à laquelle est associée une unité d'information.

Nous considérons comme unité d'information, les primitives géométriques ainsi que les relations géométriques, définies dans le chapitre 3.

- Lorsque cette unité d'information est une instance de primitive géométrique, le système considère que l'existence de l'entité géométrique désignée par l'hypothèse n'est pas sûre, et qu'elle peut être remise en cause par la suite.
- Lorsque cette unité d'information est une instance de relation géométrique, le système de perception considère que la relation peut être valide à un moment donné, mais que cette validité pourra être remise en question.

**exemple:**

Supposons qu'un capteur à ultra-son détecte la présence d'une surface. Le système crée alors une hypothèse et en étiquette la surface, pour indiquer que cette dernière est une information incertaine. Si après traitement des mesures provenant des autres capteurs cette information n'est pas confirmée, alors le système efface cette information de la mémoire.

Définissons formellement la structure d'une hypothèse.

#### 4.3.2 Mesure de confiance

Afin de caractériser la confiance que le système accorde à une hypothèse, on associe à cette dernière, un facteur de confiance numérique  $CF$ .  $CF$  prend une valeur entière positive. Cette valeur est d'autant plus grande, que la confiance accordée à l'hypothèse est forte. Nous examinerons de façon détaillée au chapitre 5, la façon dont le processus d'intégration génère les hypothèses. Disons pour l'instant, que le facteur de confiance initial d'une hypothèse dépend essentiellement de deux aspects:

1. de la fiabilité du capteur, qui a permis de fournir l'unité d'information désignée par l'hypothèse.
2. du facteur de confiance des hypothèses antérieures, à partir desquelles l'hypothèse a été créée.

Nous définissons deux valeurs extrêmes,  $CF_{min}$  et  $CF_{max}$ , en dehors desquelles l'hypothèse est traitée de façon particulière:

$CF_{min}$ : lorsque le facteur de confiance d'une hypothèse est inférieur à  $CF_{min}$ , le système considère que l'hypothèse n'est plus crédible. Aussi l'hypothèse est détruite.

$CF_{max}$ : lorsque le facteur de confiance d'une hypothèse est supérieur à  $CF_{max}$ , l'hypothèse est validée. Le système considère que l'hypothèse est suffisamment sûre, pour ne plus avoir à y revenir.

Les opérations de validation et de destruction d'une hypothèse en affecte d'autres, afin d'assurer la cohérence de la base d'hypothèses. Nous verrons dans la section 4.4 la réalisation de ces mécanismes.

Avant d'aborder ces mécanismes, il nous reste à définir les relations d'incompatibilité et de dépendance entre plusieurs hypothèses.

#### 4.3.3 Relations d'incompatibilité

La relation d'incompatibilité permet de traduire le fait que deux hypothèses ne peuvent être vraies en même temps. (Cette relation est enregistrée dans la structure, à l'aide d'un attribut qui pointe vers la liste des hypothèses qui lui sont incompatibles.) A l'inverse des travaux de J.Doyle [Doy77] ou de tous les travaux utilisant les techniques de maintenance de la vérité (TMS) dans un cadre logique, les relations d'incompatibilité doivent être explicites car les variables peuvent posséder deux estimations différentes, sans que ces dernières soient incompatibles.

#### 4.3.4 Relations de dépendance

Comme nous l'avons vu plus haut, le système de perception peut construire de nouvelles interprétations, à partir d'interprétations antérieures. Si au cours du processus d'intégration, les premières interprétations venaient à disparaître, les conséquences de ces interprétations devraient également être détruites. Cette remarque est la principale justification à l'introduction des relations de dépendance.

Au niveau structurel, l'hypothèse possède un attribut qui pointe vers une liste d'hypothèses mères dont elle dépend, ainsi qu'un attribut pointant vers une liste d'hypothèses filles, qui en dérivent.

La figure 4.1 décrit la structure d'une hypothèse et ses différents attributs.

Maintenant que nous avons défini notre structure de représentation de l'incertitude, nous devons expliciter la façon dont le système gère la base d'hypothèses.

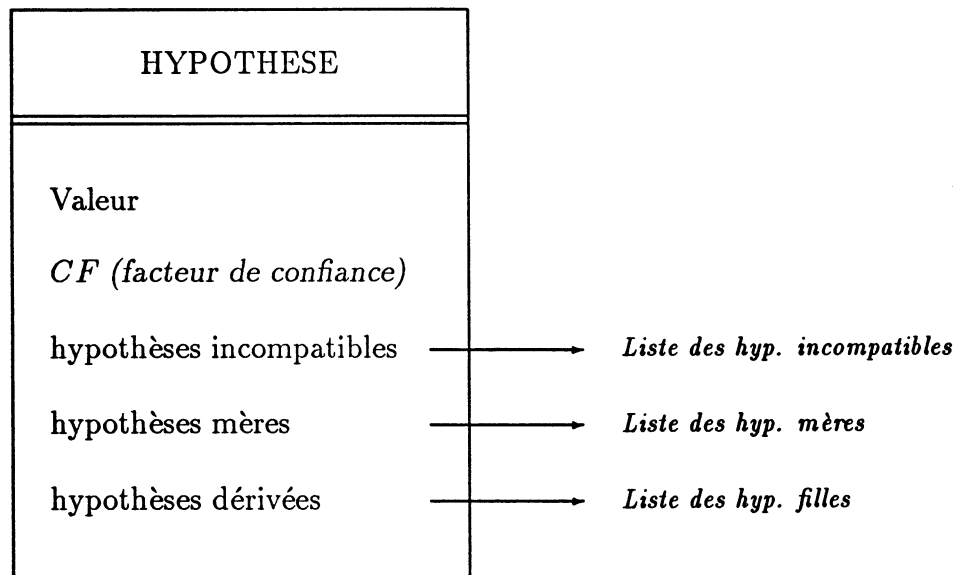


Figure 4.1: Structure d'une hypothèse

## 4.4 Gestion des incertitudes

Les capteurs fournissent des mesures sur le monde réel. Parfois, ces mesures véhiculent de l'information que le système de perception détient déjà. Nous disons alors que la mesure **confirme** l'information. Inversement, il arrive que certaines mesures contredisent l'information du système. Nous disons alors que l'information est **infirmée**. Dans ce cas, il paraît naturel que l'information concernée subisse des modifications, ou soit même remise en question.

Nous n'aborderons pas dans le détail, la définition des opérations de confirmation et de d'infirmerie d'hypothèses, car ce sont des opérations propres au processus de perception.

Dans ce chapitre, nous étudions les effets indirects qu'engendrent la modification d'une hypothèse, sur les autres hypothèses.

### 4.4.1 Modifications d'une hypothèse

Dans la figure 4.2, nous représentons graphiquement les hypothèses par des cercles, les relations de dépendance par des arcs orientés de l'hypothèse mère vers l'hypo-

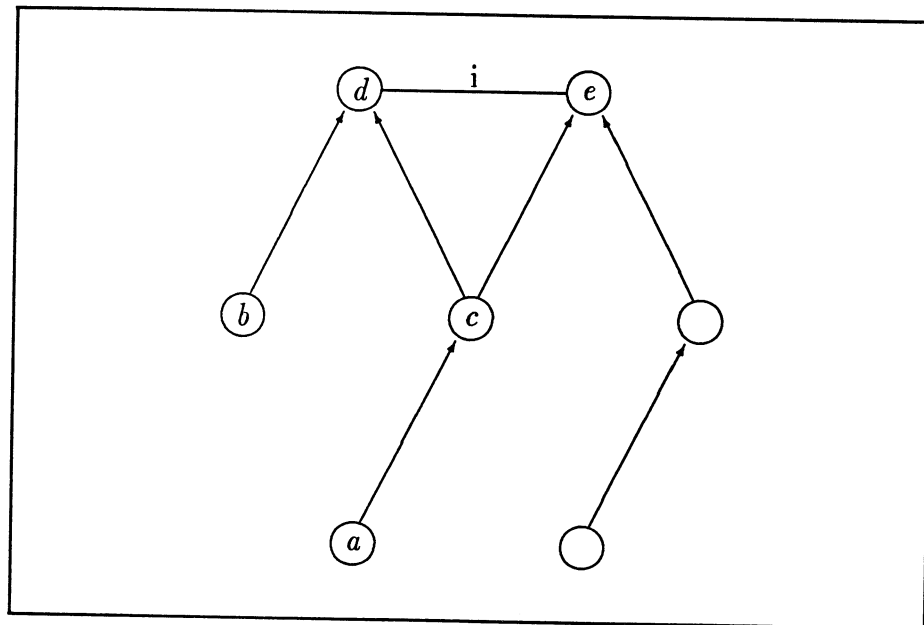


Figure 4.2: Réseau de dépendance des hypothèses

thèse fille, les relations d'incompatibilité par des arcs étiquetés "i".

Une hypothèse peut être affectée par les trois mécanismes suivants:

1. la valeur de l'hypothèse est modifiée: rappelons que l'attribut "valeur" d'une hypothèse pointe vers l'unité d'information qu'elle désigne. Dans ce cas, les hypothèses qui dérivent de cette hypothèse doivent être à nouveau réévaluées, afin de vérifier qu'elles sont toujours valides. Dans le cas où une hypothèse dérivée ne s'avèrerait plus valide, le système de gestion des hypothèses devrait la détruire.
2. L'hypothèse est détruite: dans ce cas, les hypothèses qui en dérivent doivent être détruites à leur tour. L'opération de destruction doit se propager récursivement le long des relations de dépendance.
3. l'hypothèse est validée: l'élément d'information désigné par l'hypothèse n'est plus considéré par le système comme étant plausible, mais comme étant effective. Pour maintenir la cohérence de sa base d'hypothèses, le système doit détruire les hypothèses qui sont incompatibles aux hypothèses qui ont été validées.



```
Algorithme de réévaluation d'une hypothèse  
  
Proc reevaluate  
    (parametre d'entree: h)  
debut proc  
Var:  $l_h, h_d$   
    Tester la validité de  $h$   
    Si  $h$  est valide  
        Alors  
            Faire  
                recalculer la valeur de  $h$   
                 $l_h \leftarrow$  ensemble des hypothèses filles de  $h$   
                Pour chaque hypothèse  $h_d$  de  $l_h$   
                    reevaluate( $h_d$ )  
                Fin pour  
            Fin faire  
        Sinon  
            destruction( $h$ )  
fin proc
```

Figure 4.3: Algorithme de réévaluation d'une hypothèse

Les opérations de réévaluation et de destruction d'hypothèse sont maintenant abordées.

#### 4.4.2 réévaluation d'une hypothèse

Nous décrivons figure 4.3 l'algorithme, de réévaluation d'une hypothèse.

#### Remarques

Le test de validité d'une hypothèse et le calcul de sa valeur dépendent de l'application. Dans le contexte de notre processus de perception, le test de validité est un test de mise en correspondance des valeurs des hypothèses mères.

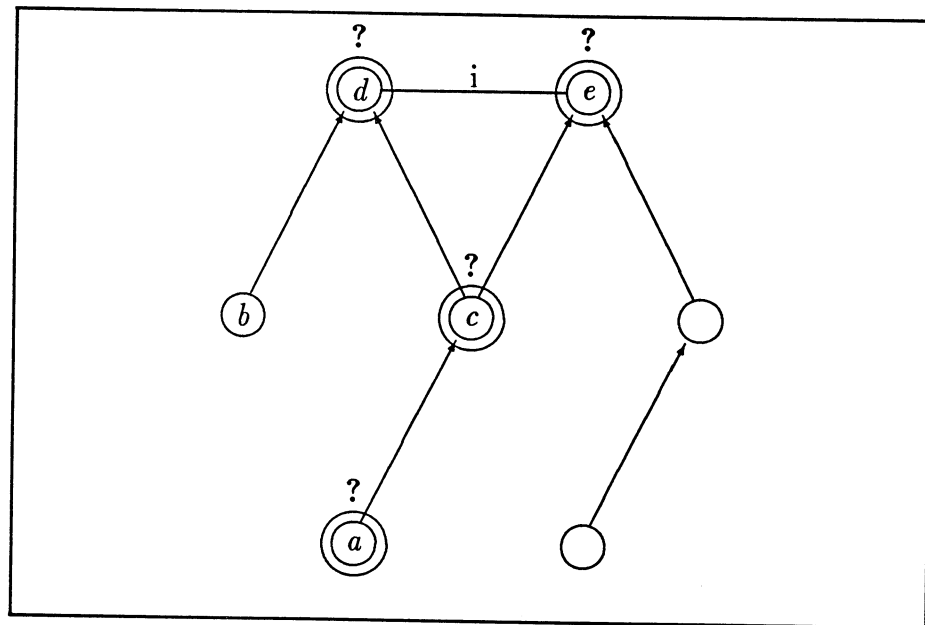


Figure 4.4: réévaluation d'une hypothèse

Le calcul de la valeur est une fusion des valeurs des hypothèses mères.

Pour illustrer le mécanisme de réévaluation, reprenons l'exemple de la figure 4.2

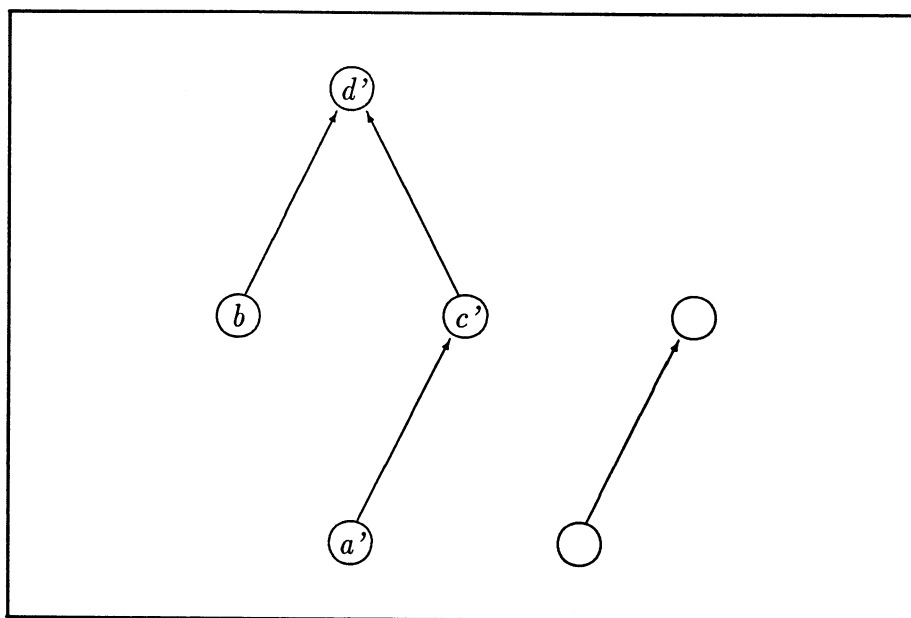
Supposons que l'hypothèse  $a$  doive être réévaluée. Si elle reste valide, on doit recalculer sa valeur et réévaluer sa dérivée directe  $b$  ainsi que ses dérivées indirectes  $d$  et  $f$ . Nous avons représenté les hypothèses à réévaluer en les entourant d'un cercle, et en les étiquetant d'un "?".

Si les hypothèses  $c$  et  $d$  restent valides, mais que l'hypothèse  $e$  est remise en cause, la situation finale est celle décrite par la figure 4.5

La notation  $a'$  signifie que la valeur de l'hypothèse  $a$  a été modifiée.

#### 4.4.3 Destruction d'une hypothèse

Nous décrivons figure 4.6 l'algorithme de destruction d'une hypothèse.

Figure 4.5: Situation après réévaluation de  $a$ 

### Remarques

L'opération de destruction d'une hypothèse consiste simplement à détruire récursivement toutes ses hypothèses dérivées et mettre à jour les relations de dépendance, ainsi que les relations d'incompatibilité.

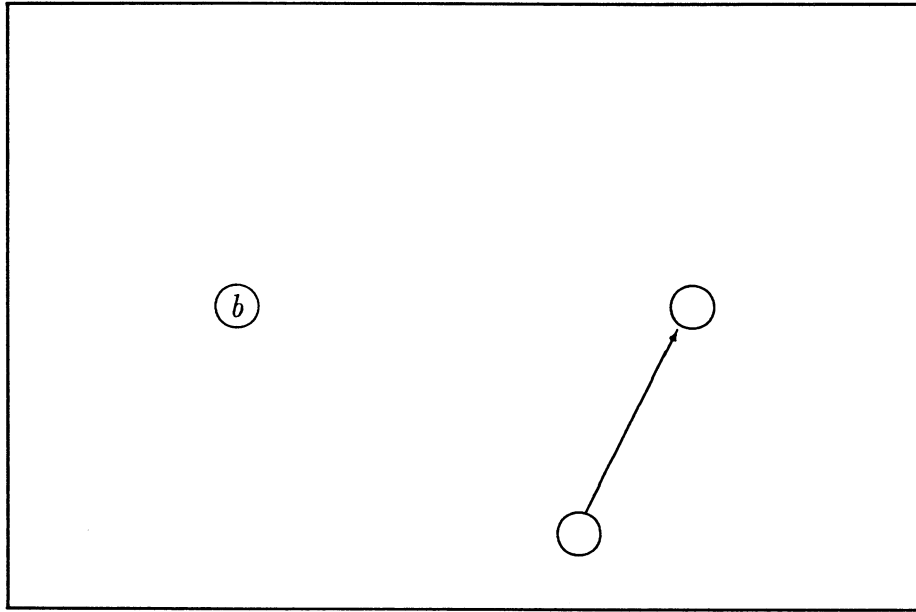
Là encore, la décision de détruire une hypothèse, dépend essentiellement de l'application. Par exemple, dans notre contexte de processus de perception, nous détruisons les hypothèses dont le facteur de confiance est inférieur au seuil critique  $CF_{min}$ .

Pour illustrer le mécanisme de destruction d'une hypothèse, reprenons l'exemple introduit dans la section 4.3.4, dans l'état où nous l'avons laissé. Cet état est représenté figure 4.5.

Supposons que l'hypothèse  $a'$  doive être détruite. L'algorithme de destruction détruit sa dérivée directe  $c'$ , ainsi que sa dérivée indirecte  $d'$ . L'état final est représenté figure 4.7.

**Algorithme de destruction d'une hypothèse****Proc** destruction*(parametre d'entree:  $h$ )***debut proc****Var:**  $l_{hd}$ ,  $l_{hi}$ ,  $h_d$ ,  $h_i$  $l_{hd} \leftarrow$  ensemble des hypothèses filles de  $h$  $l_{hi} \leftarrow$  ensemble des hypothèses incompatibles avec  $h$ **Pour** chaque hypothèse  $h_d$  de  $l_{hd}$ **Faire**détruire( $h_d$ )détruire les relations de dépendance entre  $h$  et  $h_d$ **Fin faire****Fin pour****Pour** chaque hypothèse  $h_i$  de  $l_{hi}$ détruire la relation d'incompatibilité entre  $h$  et  $h_i$ **Fin pour****fin proc**

Figure 4.6: Algorithme de destruction d'une hypothèse

Figure 4.7: Situation après destruction de  $a'$ 

## 4.5 Résumé

Nous avons introduit un formalisme qui permet de décrire l'incertitude d'informations.

Ce formalisme est basé sur la définition d'un réseau de dépendances, dans lequel chaque noeud est une hypothèse.

Une hypothèse est une structure qui permet de désigner un élément d'information. Un attribut  $CF$  quantifie la confiance que le système accorde à la validité de cet élément d'information.

Les arcs du réseau décrivent deux types de relations:

les relations de dépendance qui traduisent la dépendance d'un élément d'information sur un autre.

les relations d'incompatibilité relient les éléments d'information qui ne peuvent être vrais simultanément.

Le réseau de dépendance permet de définir la confiance que le système de perception attribue aux primitives et relations géométriques, ainsi que les interdépendances de ces informations.

---

Nous avons défini un module permettant de gérer la cohérence de ce réseau de dépendance, lorsque se produisent les événements externes tels que la validation d'une hypothèse, la destruction d'une hypothèse ou la modification de l'information désignée par l'hypothèse.



## Chapitre 5

# Intégration sensorielle

### 5.1 Introduction

Dans les chapitres 2 et 3, nous avons introduit notre mode de représentation des informations. Ce mode de représentation nous sert tout aussi bien à décrire les informations récentes provenant des capteurs (*informations sensorielles*), que les informations stockées par le système dans le modèle composite. Dans le chapitre 4, nous avons introduit un cadre formel pour permettre au système de perception de prendre en compte des informations incertaines et éventuellement incompatibles.

Dans ce chapitre, nous expliquons comment en pratique, le système de perception intègre les informations sensorielles pour produire et mettre à jour une description géométrique robuste, cohérente et précise de son environnement.

Nous aborderons quatre points:



- lorsqu'un capteur fournit de nouveaux éléments d'informations au système, il est nécessaire de déterminer l'ensemble des éléments d'informations antérieurs, qui sont susceptibles d'être affectés par les nouvelles informations. Lorsqu'un nouvel élément d'information affecte un ancien élément d'information, nous disons que ces deux éléments sont en correspondance. Dans la première section, nous définissons un **ensemble de tests de correspondance locale**.
- lorsque deux éléments d'information ont été mis en correspondance locale, on peut les fusionner pour construire une information qui contiendrait un consensus des informations contenues dans chacun d'entre eux. Dans la seconde section, nous définissons un **ensemble de procédures de fusion**.
- la plupart du temps, les capteurs fournissent un flux important d'informations. Le système dispose lui-même d'une quantité d'informations conséquente. Plutôt que de tester la correspondance locale entre tous les nouveaux éléments d'information et tous les anciens éléments d'information de façon systématique, nous devons utiliser convenablement la structure de notre base d'informations et définir une stratégie de contrôle permettant d'effectuer une mise en correspondance efficace, c'est à dire, une mise en correspondance, qui évite à tout prix l'écueil d'une combinatoire élevée. Dans la troisième section nous définissons la **stratégie de mise en correspondance globale**.
- le système traite les éléments mis en correspondance pour affiner les informations contenues dans le modèle, pour enrichir le modèle et pour confirmer ou infirmer les informations incertaines qu'il possède. Ce mécanisme, que nous appelons processus d'intégration, s'effectue dynamiquement et cycliquement. La dernière section décrit l'algorithme d'exécution d'un cycle d'intégration.

## 5.2 Mise en correspondance locale de primitives

Dans cette section nous définissons les opérations de mise en correspondance des primitives géométriques.

Notations:

Les segments 3D sont notés  $S_a$  et  $S_b$ , les éléments de surface sont notés  $E_a$  et  $E_b$ . Le segment  $S_a$  est défini par son point milieu  $P_a$ , sa direction  $N_a$ , sa longueur  $L_a$ , à  $P_a$  et  $N_a$  sont associées les matrices de covariance  $\Lambda_{P_a}$  et  $\Lambda_{N_a}$ .

L'élément de surface  $E_a$  est défini par un point  $P_a$ , par sa normale  $N_a$ .

$0_i$  représente le vecteur colonne nul de dimension  $i$ .

$I_i$  représente la matrice identité de dimension  $i \times i$ .

### 5.2.1 mise en correspondance de deux segments-3D

Les tests que nous définissons dans ce chapitre, tiennent compte de l'imprécision des paramètres mis en jeu. Pour cela, on part d'une formulation ordinaire (sans tenir compte de l'imprécision), qui s'écrit en général:

$$f(X_1, X_2, \dots, X_n) = 0 \quad (5.34)$$

Puis on évalue la matrice de covariance de la variable aléatoire  $f(X_1, X_2, \dots, X_n)$  en utilisant la technique présentée à la section 2.4.3. Enfin, en utilisant la distance normalisée, définie dans la section 2.4.1, on vérifie que l'écart entre 0 et la valeur  $f(X_1, X_2, \dots, X_n)$  effectivement obtenue, peut être imputée à l'imprécision des v.a.m.  $X_1, X_2, \dots, X_n$ . Cette vérification se réalise à l'aide d'une inéquation, qui prend en compte les matrices de covariance des différentes v.a.m..

#### Test de parallélisme

La relation s'écrit:

$$N_a - N_b = 0_3 \quad (5.35)$$

le test s'écrit donc:

$$(N_a - N_b)^{tr} (\Lambda_{N_a} + \Lambda_{N_b})^{-1} (N_a - N_b) < k \quad (5.36)$$

#### Test d'alignement

Les deux segments  $S_a$  et  $S_b$  sont alignés, si les droites affines qui les supportent sont confondues. Leur parallélisme doit d'abord être vérifié. La relation supplémentaire s'écrit:

$$N_a \wedge (P_a - P_b) = 0_3 \quad (5.37)$$

où “ $\wedge$ ” dénote le produit vectoriel. Le test s’écrit:

$$(N_a \wedge (P_a - P_b))^{tr} (V_{N_a} (\Lambda_{P_a} + \Lambda_{P_b}) V_{N_a}^{tr} + V_p \Lambda_{N_a} V_p^{tr})^{-1} (N_a \wedge (P_a - P_b)) < k \quad (5.38)$$

où:

$$V_{N_a} = \begin{bmatrix} 0 & -z_{N_a} & y_{N_a} \\ z_{N_a} & 0 & -x_{N_a} \\ -y_{N_a} & x_{N_a} & 0 \end{bmatrix} \text{ et } V_p = \begin{bmatrix} 0 & -z_{P_a} + z_{P_b} & y_{P_a} - y_{P_b} \\ z_{P_a} - z_{P_b} & 0 & -x_{P_a} + x_{P_b} \\ -y_{P_a} + y_{P_b} & x_{P_a} - x_{P_b} & 0 \end{bmatrix}$$

### Test de recouvrement

Les deux segments  $S_a$  et  $S_b$  se recouvrent, s’ils sont alignés et de plus, possèdent au moins un point commun. La condition à vérifier s’écrit:

$$\exists P'_a \in S_a, \exists P'_b \in S_b | P'_a - P'_b = 0_3 \quad (5.39)$$

Pour se ramener à une relation sans quantificateur, il nous suffit de créer un point aléatoire fictif  $Q_a$  qui peut représenter *tous* les points de  $S_a$ , et un point aléatoire fictif  $Q_b$  qui peut représenter *tous* les points de  $S_b$ . La condition de recouvrement s’écrit simplement:

$$Q_a - Q_b = 0_3 \quad (5.40)$$

les points particuliers  $P'_a$  et  $P'_b$  de l’équation 5.39 sont des occurrences des points aléatoires  $Q_a$  et  $Q_b$ . Le point  $Q_a$  est construit selon la formule:

$$Q_a = P_a + \alpha N_a \quad (5.41)$$

où  $\alpha$  est une variable aléatoire dont la variance est  $\frac{L_a^2}{4}$ . Intuitivement,  $Q_a$  est obtenu en “faisant glisser” aléatoirement le point  $P_a$  le long de la direction  $N_a$  jusqu’aux extrémités du segment  $S_a$ .

La matrice de covariance de  $Q_a$  est obtenue en considérant dans l’équation 5.41,  $Q_a$  comme étant une fonction de  $(P_a, \alpha, N_a)$ , puis en utilisant la technique de transformation développée dans le paragraphe 2.4.3. On obtient alors:

$$\Lambda_{Q_a} = \begin{bmatrix} I_3 & N_a & I_3 \end{bmatrix} \begin{bmatrix} \Lambda_{P_a} & 0_3 & 0_3 \\ 0_3^{tr} & \frac{L_a^2}{4} & 0_3^{tr} \\ 0_3 & 0_3 & a^2 \Lambda_{N_a} \end{bmatrix} \begin{bmatrix} I_3 \\ N_a^{tr} \\ I_3 \end{bmatrix} \quad (5.42)$$

le test de recouvrement s'écrit:

$$(Q_a - Q_b)^{tr} (\Lambda_{Q_a} + \Lambda_{Q_b})^{-1} (Q_a - Q_b) < k \quad (5.43)$$

### 5.2.2 Mise en correspondance d'un segment-3D avec un élément de surface

#### Test de parallélisme

Le segment  $S_a$  et l'élément de surface  $E_b$  sont parallèles si:

$$N_a \cdot N_b = 0_3 \quad (5.44)$$

où “.” dénote le produit scalaire. Le test s'écrit donc:

$$\frac{(N_a \cdot N_b)^2}{\sigma^2} < k \quad (5.45)$$

où:

$$\sigma^2 = \begin{bmatrix} N_b^{tr} & N_a^{tr} \end{bmatrix} \begin{bmatrix} \Lambda_{N_a} & 0_3 \\ 0_3^{tr} & \Lambda_{N_b} \end{bmatrix} \begin{bmatrix} N_b \\ N_a \end{bmatrix}$$

#### Test d'appartenance

Le segment  $S_a$  appartient à l'élément de surface  $E_b$ , si  $S_a$  est parallèle à  $E_b$  et si de plus le point  $P_a$  de  $S_a$  appartient à l'élément de surface  $E_b$ . Soit la condition:

$$\overrightarrow{P_a P_b} \cdot N_b = 0_3 \quad (5.46)$$

Le test d'appartenance s'écrit alors:

$$\frac{(\overrightarrow{P_a P_b} \cdot N_b)^2}{N_b^{tr} (\Lambda_{P_a} + \Lambda_{P_b}) N_b + \overrightarrow{P_a P_b}^{tr} \Lambda_{N_b} \overrightarrow{P_a P_b}} < k \quad (5.47)$$

### 5.2.3 Mise en correspondance de deux éléments de surface

deux éléments de surface  $S_a$  et  $S_b$  sont en correspondance si

$$N_a \wedge N_b = 0_3 \quad (5.48)$$

Le test s'écrit alors:

$$(N_a \wedge N_b)^{tr} (V_{N_b} \Lambda_{N_a} V_{N_b}^{tr} + V_{N_a} \Lambda_{N_b} V_{N_a}^{tr}) (N_a \wedge N_b) \quad (5.49)$$

où:

$$V_{N_a} = \begin{bmatrix} 0 & -z_{N_a} & y_{N_a} \\ z_{N_a} & 0 & -x_{N_a} \\ -y_{N_a} & x_{N_a} & 0 \end{bmatrix}$$

et où

$$V_{N_b} = \begin{bmatrix} 0 & -z_{N_b} & y_{N_b} \\ z_{N_b} & 0 & -x_{N_b} \\ -y_{N_b} & x_{N_b} & 0 \end{bmatrix}$$

### 5.3 Fusion de primitives

#### Principe

Les primitives géométriques constituent des estimateurs d'entités géométriques réelles. La fusion de deux primitives géométriques est une opération que l'on peut réaliser, lorsque les deux entités qu'elles estiment sont reliées géométriquement, par exemple lorsque deux segments 3D sont parallèles. On peut alors utiliser la contrainte induite par la direction d'un des segments et la relation de parallélisme, pour contraindre la direction de l'autre segment et améliorer son estimation antérieure.

Plus généralement, la fusion de deux primitives géométriques consiste à réévaluer les paramètres numériques d'une des primitives, en tenant compte des relations géométriques liant ces deux primitives, et leurs paramètres numériques initiaux.

#### 5.3.1 Fusion de deux sommets

Soient  $P_a$  et  $P_b$ , deux estimateurs d'un même sommet<sup>1</sup> physique  $P$ . Leur fusion permet d'obtenir une meilleure estimation de  $P$ , sous la forme d'un nouveau sommet  $P^*$ .

Notations:

$P_a$  et  $P_b$  sont caractérisés par leurs coordonnées  $(x_a, y_a, z_a)$  et  $(x_b, y_b, z_b)$ , et par les

---

<sup>1</sup>Rappel: Un sommet est une structure géométrique qui permet de regrouper un ensemble de segments 3D qui concourent à l'une de leurs extrémités. Le sommet est caractérisé par le point 3D sur lequel concourent ces segments

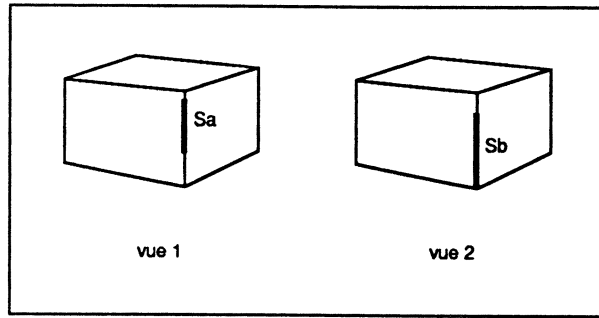


Figure 5.1: segments 3D se correspondant

matrices de covariances  $3 \times 3$   $\Lambda_{P_a}$  et  $\Lambda_{P_b}$ , qui décrivent la précision de l'estimation de ces coordonnées.

$P^*$  et  $\Lambda_{P^*}$  sont obtenues à partir de la fonction de combinaison de v.a.m. introduite au paragraphe 2.4.2. Soit:

$$(P^*, \Lambda_{P^*}) = \text{combine}(P_a, \Lambda_{P_a}, P_b, \Lambda_{P_b}) \quad (5.50)$$

### 5.3.2 Fusion de deux segments-3D

Soient  $S_a$  et  $S_b$  deux estimateurs d'un même segment physique  $S$ , leur fusion permet d'obtenir une meilleure estimation de  $S$ , sous la forme d'un segment 3D  $S^*$ .

Notations:

les extrémités de  $S_a$  et  $S_b$  sont notées  $P_{a1}$ ,  $P_{b1}$ ,  $P_{a2}$  et  $P_{b2}$ .

Les techniques d'inférences qui ont permis d'extraire les estimateurs  $S_a$ , et  $S_b$  du segment physique  $S$  ont pu tronquer ces estimateurs (voir figure 5.1). Par conséquent, les points  $P_{a1}$  et  $P_{b1}$  peuvent ne pas correspondre au même point physique. Il en est de même pour les points  $P_{a2}$  et  $P_{b2}$ . Cet artefact rend l'estimation de  $S^*$  un peu plus complexe qu'une application directe sur les extrémités des segments  $S_a$  et  $S_b$ , de la technique de combinaison de vam.

Dans un premier temps, on réestime la position des points  $P_{a1}$ ,  $P_{b1}$ ,  $P_{a2}$  et  $P_{b2}$ , en tenant compte du fait que  $P_{a1}$  et  $P_{a2}$ , appartiennent à la droite physique estimée par la droite  $(P_{b1}, P_{b2})$ .

Soit le point  $Q_b$  défini par:

$$Q_b = P_b + \alpha_b N_b \quad (5.51)$$

où  $\alpha_b$  est une variable aléatoire dont la variance est  $\sigma_{\alpha_b}^2 = (\frac{L_b}{2} + L_a)^2$ .  $Q_b$  est un point aléatoire, qui représente *tous* les points de la droite  $(P_{b1}, P_{b2})$ , situés à une distance maximale de  $\frac{L_b}{2} + L_a$  par rapport à  $P_b$ .

La matrice de covariance de  $Q_b$  est obtenue en considérant dans l'équation 5.51,  $Q_b$  comme étant une fonction de  $(P_b, \alpha_b)$ , puis en utilisant la technique de transformation développée dans le paragraphe 2.4.3. On obtient alors:

$$\Lambda_{Q_b} = \begin{bmatrix} I_3 & N_b & I_3 \end{bmatrix} \begin{bmatrix} \Lambda_{P_b} & 0_3 & 0_3 \\ 0_3^{tr} & \sigma_{\alpha_b}^2 & 0_3^{tr} \\ 0_3 & 0_3 & a^2 \Lambda_{N_b} \end{bmatrix} \begin{bmatrix} I_3 \\ N_b^{tr} I_3 \end{bmatrix} \quad (5.52)$$

Déclarer que :le point physique estimé par  $P_{a1}$  est situé sur la droite physique estimée par  $(P_{b1}, P_{b2})$ , revient à dire que  $P_{a1}$  et  $Q_b$  sont deux estimateurs d'un même point physique. On peut alors former un meilleur estimateur  $P_{a1}^*$  de ce point, en combinant ces deux estimations à l'aide de l'opération définie au paragraphe 2.4.2:

$$(P_{a1}^*, \Lambda_{P_{a1}^*}) = combine(P_{a1}, \Lambda_{P_{a1}}, Q_b, \Lambda_{Q_b}, N_b, \Lambda_{N_b}) \quad (5.53)$$

Le même raisonnement appliqué à  $P_{a2}$ , fournit un meilleur estimateur  $P_{a2}^*$ .

Lorsqu'on inverse les rôles des segments 3D  $S_a$  et  $S_b$ , et que l'on applique le raisonnement à  $P_{b1}$  et  $P_{b2}$ , on obtient les nouveaux estimateurs  $P_{b1}^*$  et  $P_{b2}^*$ . Les segments  $S_a$ ,  $S_b$  et les différents points créés sont représentés sur la figure 5.2 Pour construire le segment  $S^*$  optimal, nous choisissons ses deux extrémités parmi les quatre points  $P_{a1}^*$ ,  $P_{a2}^*$ ,  $P_{b1}^*$ ,  $P_{b2}^*$ , de telle façon que sa longueur soit la plus grande possible. Les extrémités  $P_1^*$  et  $P_2^*$  de  $S$  sont obtenues en appliquant aux quatres points l'algorithme décrit figure 5.3

Nous avons défini le segment 3D  $S^*$ , par ses deux extrémités  $P_1^*$  et  $P_2^*$ . Les autres paramètres de sa représentation sont donnés par les formules:

$$P^* = \frac{P_1^* + P_2^*}{2}$$

$$L^* = \|\overline{P_1^* P_2^*}\|$$

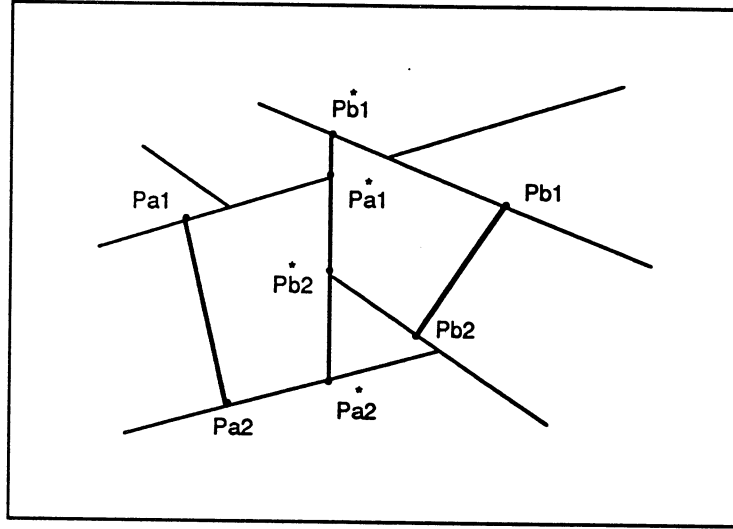


Figure 5.2: Fusion de deux segments 3D: construction des extrémités

$$N^* = \frac{\overrightarrow{P_1^* P_2^*}}{\| \overrightarrow{P_1^* P_2^*} \|}$$

$$\Lambda_{P^*} = \frac{1}{4}(\Lambda_{P_1^*} + \Lambda_{P_2^*})$$

$$\Lambda_{N^*} = \frac{\Lambda_{P_1^*} + \Lambda_{P_2^*}}{\| \overrightarrow{P_1^* P_2^*} \|^2}$$

Attention, ces formules ne peuvent s'utiliser que si les précisions des différents paramètres sont indépendantes les unes des autres

### 5.3.3 Fusion de deux éléments de surface

Soient  $E_a$  et  $E_b$ , deux éléments de surface estimant la même surface physique. Leur fusion permet de produire une meilleure estimation sous la forme d'un nouvel élément de surface  $E^*$ .

Notations:

$E_a$  et  $E_b$  sont définis par leur normale  $N_a$  et  $N_b$ , par un point  $P_a$  et  $P_b$ , et par les matrices de covariance de chacun de ces paramètres  $\Lambda_{N_a}$ ,  $\Lambda_{N_b}$ ,  $\Lambda_{P_a}$  et  $\Lambda_{P_b}$ .

La normale de  $E^*$  est obtenue en combinant les normales de  $E_a$  et  $E_b$ .

$$(N^*, \Lambda_{N^*}) = \text{combine}(N_a, \Lambda_{N_a}, N_b, \Lambda_{N_b}) \quad (5.54)$$



**Algorithme de choix des extrémités du segment 3D**

```

Si  $|P_{a1}^* P_{b1}^*| < |P_{a1}^* P_{a2}^*|$ 
  Si  $\vec{P_{a1}^* P_{b1}^*} \cdot \vec{P_{a1}^* P_{a2}^*} > 0$ 
    Si  $|P_{a1}^* P_{b2}^*| < |P_{a1}^* P_{a2}^*|$ 
      Si  $\vec{P_{a1}^* P_{b2}^*} \cdot \vec{P_{a1}^* P_{a2}^*} > 0$ 
         $P_1^* = P_{a1}^*$  et  $P_2^* = P_{a2}^*$ 
      Sinon
         $P_1^* = P_{b2}^*$  et  $P_2^* = P_{a2}^*$ 
    Sinon
      Si  $\vec{P_{a1}^* P_{b2}^*} \cdot \vec{P_{a1}^* P_{a2}^*} > 0$ 
         $P_1^* = P_{a1}^*$  et  $P_2^* = P_{b2}^*$ 
      Sinon
         $P_1^* = P_{b2}^*$  et  $P_2^* = P_{a2}^*$ 
  Sinon
    Si  $|P_{b1}^* P_{b2}^*| < |P_{b1}^* P_{a2}^*|$ 
      Si  $\vec{P_{b1}^* P_{b2}^*} \cdot \vec{P_{b1}^* P_{a2}^*} > 0$ 
         $P_1^* = P_{b1}^*$  et  $P_2^* = P_{a2}^*$ 
      Sinon
         $P_1^* = P_{b2}^*$  et  $P_2^* = P_{a2}^*$ 
    Sinon
      Si  $\vec{P_{b1}^* P_{b2}^*} \cdot \vec{P_{b1}^* P_{a2}^*} > 0$ 
         $P_1^* = P_{b1}^*$  et  $P_2^* = P_{b2}^*$ 
      Sinon
         $P_1^* = P_{b2}^*$  et  $P_2^* = P_{a2}^*$ 

```

Figure 5.3: Algorithme de fusion de deux segments 3D

Il nous reste à définir la position de  $E^*$ .

Nous définissons le point  $P^*$  par lequel passe l'élément de surface  $E^*$ , comme le nouvel estimateur de  $P_a$ , connaissant la relation supplémentaire:

$$N^* \wedge (P_a - P_b) = 0 \quad (5.55)$$

(Cette relation exprime simplement que les points  $P_a$  et  $P_b$  sont sur un plan perpendiculaire au vecteur  $N^*$ .)  $P^*$  et sa covariance sont alors calculés en utilisant la technique du filtre de Kalman étendu, présenté à la section 2.4.2.

#### 5.3.4 Fusion d'un élément de surface et d'un segment-3D

Soient  $S_a$  un segment 3D et  $E_b$ , un élément de surface. Supposons que l'on ait pu établir que le segment physique estimé par  $S_a$  appartenait au plan physique estimé par  $E_b$ . Leur fusion permet de fournir un meilleur estimateur du segment physique sous la forme d'un segment 3D  $S_a^*$ , ainsi qu'un meilleur estimateur du plan physique sous la forme d'un élément de surface  $E_b^*$ .

Ces deux estimateurs sont calculés à partir des relations:

$$N_a \cdot N_b = 0_3 \quad (5.56)$$

et

$$N_b \cdot (P_a - P_b) = 0_3 \quad (5.57)$$

La relation 5.56 exprime que  $S_a$  est parallèle à  $E_b$ .

La relation 5.57 exprime que  $P_a$  est un point de  $E_b$ .

Le vecteur directeur  $N_a^*$  de  $S_a^*$  est calculé comme un nouvel estimateur de  $N_a$ , connaissant l'équation 5.56.

Le vecteur normal  $N_b^*$  de  $E_b^*$  est calculé comme un nouvel estimateur de  $N_b$ , connaissant la même équation.

Les points  $P_a^*$  et  $P_b^*$  sont calculés simultanément, comme des nouveaux estimateurs de  $P_a$  et  $P_b$ , connaissant l'équation 5.57.

## 5.4 Stratégie de mise en correspondance globale

Le problème que nous traitons ici, est celui qui consiste à mettre en correspondance deux ensembles de primitives géométriques. Ces deux ensembles peuvent être, par exemple, l'ensemble des primitives du modèle composite, et l'ensemble des primitives générés par les observations d'un capteur.

Notons

$$E = \{e_i \mid i \in \{1, \dots, n\}\} \text{ et } F = \{f_j \mid j \in \{1, \dots, m\}\}$$

les deux ensembles de primitives que l'on veut mettre en correspondance.

La première solution qui vient à l'esprit, consiste à tester exhaustivement la correspondance entre chaque élément de l'ensemble  $E$  et chaque élément de l'ensemble  $F$ , avec les fonctions de test que nous avons défini dans la section 5.2. Rappelons que ces tests permettent de détecter la possibilité pour deux primitives géométriques de représenter deux estimateurs cohérents de la même entité géométrique, ou d'être reliées par une relation géométrique. Par exemple, si l'on teste le parallélisme de deux segments-3D, et que le résultat est positif, le système en conclut que les deux segments peuvent être parallèles, et non que les deux segments sont parallèles. Cette information est prise en compte dans le système par le biais d'une nouvelle hypothèse, qui désigne la relation de parallélisme, et permet ainsi d'enregistrer la plausibilité de la relation.

Ainsi, pour chaque mise en correspondance  $(e_i, f_j)$  réussie, le système génère une hypothèse désignant l'élément d'information qui résulterait de la validation de l'hypothèse.

La complexité de cet algorithme est exponentielle en  $n^m$ . De toute évidence, cette complexité est rédhibitoire, dès que l'un des deux ensembles contient un nombre important de primitives. La combinatoire de cet algorithme est élevée pour deux raisons:

- pour chaque élément  $e_i$  de l'ensemble  $E$ , on cherche ses correspondants dans l'ensemble  $F$  tout entier, sans faire de sélection préalable parmi ses éléments  $f_j$ .

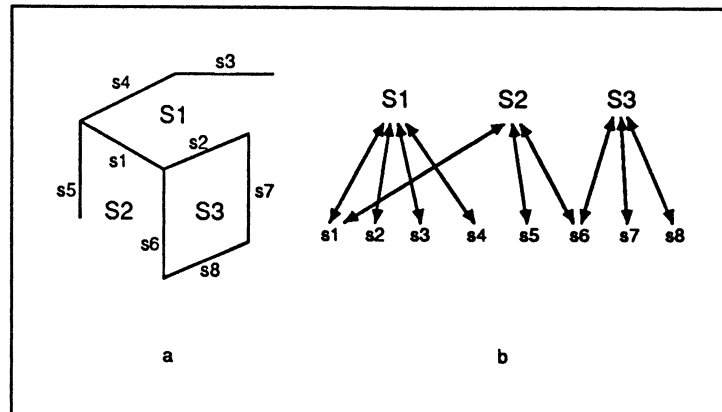


Figure 5.4: primitives 3D du modèle

- les éléments  $e_i$  de  $E$  dont on veut trouver un correspondant dans  $F$ , sont pris dans un ordre arbitraire. Or, pour certains de ces éléments, il est plus facile de trouver un correspondant, parce que les tests mis en jeu dans la recherche du correspondant sont moins coûteux en temps de calcul, ou parce que le nombre de correspondants possibles dans l'ensemble  $F$  est faible.

L'aspect hiérarchique de notre représentation des données géométriques va nous permettre de résoudre ces deux problèmes. Rappelons que selon notre représentation, les informations géométriques de l'environnement sont structurées en un graphe, dans lequel les noeuds sont constitués de primitives géométriques simples, tels que des sommets, des segments-3D, des facettes planes.... Les arcs de ce graphe représentent les relations géométriques qui peuvent exister entre les primitives géométriques. Les relations géométriques comprennent les relations de parallélisme, les relations d'appartenance,...

Considérons les primitives-3D représentées figure 5.4 a. Leur description symbolique sous forme de graphe est indiquée figure 5.4 b. Supposons que l'ensemble de ces primitives constitue le modèle composite, et que les données observées soient formées des primitives-3D représentées figure 5.5 a, ces primitives formant le graphe indiqué figure 5.5 b. L'algorithme sélectionne les couples  $(e_i, f_j)$  à mettre en correspondance en utilisant deux critères:

**Choix de  $e_i$ :** Le système prend l'élément le plus abstrait, c'est à dire le plus haut dans la hiérarchie de la représentation. Dans notre exemple, nous choisissons

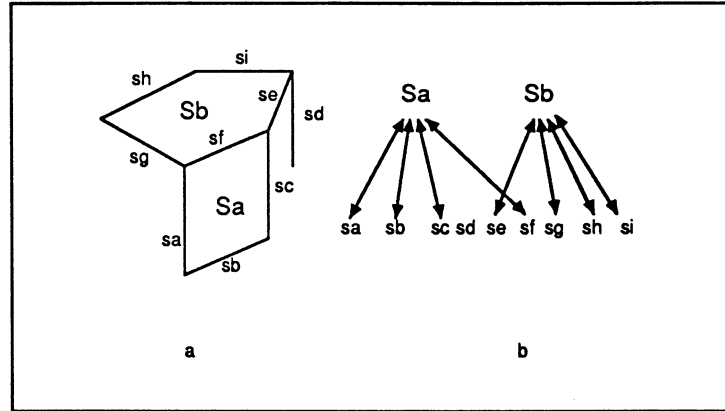


Figure 5.5: primitives 3D observées

de chercher une correspondance pour l'élément de surface  $S_1$ . L'ensemble  $F$  ne contenant que 2 éléments de surface, la recherche des correspondants possibles de  $S_1$  sera rapide.

**Choix de  $f_i$ :** Le système cherche le correspondant de  $S_1$  en commençant par l'élément le plus abstrait de  $F$ . Dans notre exemple, l'ensemble  $F$  ne contient qu'une surface  $S_a$ . Le système teste donc la correspondance entre  $S_1$  et  $S_a$ . Le résultat du test se révélant négatif, le système passe à l'élément de surface de  $F$  suivant, et teste la correspondance entre  $S_1$  et  $S_b$ .

Les surfaces  $S_2$  et  $S_a$  étant en correspondance, le système construit un élément de surface, obtenu en fusionnant  $S_1$  et  $S_2$ , et génère une hypothèse désignant ce nouvel élément. Cette mise en correspondance réussie, permet de tester la correspondance entre les segments-3D de  $S_1$  et les segments-3D de  $F$ . Les segments contenus dans l'élément de surface  $S_a$  ne peuvent correspondre aux segments-3D de  $S_1$  car  $S_1$  et  $S_a$  ne se correspondent pas. Il suffit donc de chercher les correspondants potentiels des segments de  $S_1$  parmi les segments de  $S_b$ , et parmi les segments qui n'appartiennent à aucun élément de surface. Par exemple, le correspondant du segment  $s_1$  est à chercher dans l'ensemble  $\{s_e, s_f, s_g, s_h, s_i\}$ .

L'algorithme de mise en correspondance est décrit formellement figure 5.6. (Remarque: dans cet algorithme,  $lp_1$  et  $lp_2$  sont structurés de façon que les primitives de type surface et sommet sont accédées en premier, et que les primitives de type segment 3D qui ne sont reliés

**Algorithme de mise en correspondance de deux descriptions partielles**

**Proc** correspondance-globale

(parametres d'entree:  $lp_1, lp_2$ , parametre de sortie:  $l$ )

$lp_1$  et  $lp_2$  sont les deux ensembles de primitives a mettre en correspondance

**debut proc**

**Var:**  $l_1, l_2$

$l \leftarrow \emptyset$

**Pour** chaque element  $p_1$  de  $lp_1$

$l_1 \leftarrow \emptyset$

**Pour** chaque element  $p_2$  de  $lp_2$

**Faire**

correspondance-locale( $p_1, p_2, l_2$ )

ajouter  $l_2$  a  $l_1$

**Fin faire**

**Fin pour**

ajouter  $l_1$  a  $l$

**Fin pour**

**fin proc**

Figure 5.6: Algorithme de mise en correspondance globale

géométriquement à aucune autre primitive, sont accédées en dernier)

## 5.5 Algorithme d'un cycle d'intégration

### 5.5.1 Principes

Les données manipulées par le système de perception se décomposent en trois groupes:

**Les éléments sûrs** qui constituent l'information fiable du système. Cet ensemble fournit les données utilisables par les autres fonctions de l'agent.

**Algorithme de mise en correspondance locale de primitives****Proc** correspondance-locale*(parametre d'entree:  $p_1, p_2$ , parametre de sortie:  $l$ )* *$p_1$  et  $p_2$  sont les deux primitives dont on veut tester la correspondance***debut proc****Var:**  $l_1, l_2, l', h, h_1, h_2, l_{h_1}, l_{h_2}$  $l \leftarrow \emptyset$ **Si** les types de  $p_1$  et  $p_2$  sont differents**sortie proc****Si** correspond( $p_1, p_2$ ) vrai**Faire** $p' \leftarrow fusion(p_1, p_2)$ creation d'une hypothese  $h$  designant  $p'$  $h_{p_i} \leftarrow$  l'hypothese designant  $p_i$  ( $i \in 1, 2$ ) $CF_h = CF_{h_{p_1}} + 1$  $l_{h_i} \leftarrow$  l'ensemble des hypotheses filles de  $h_{p_i}$  ( $i \in 1, 2$ )**Pour** chaque element  $h_i$  de  $l_{h_i}$ **Faire**tester la correspondance entre la primitive designee par  $h_i$  et  $p'$ Si le test est negatif enregistrer l'incompatibilite de  $h$  et  $h_i$ **Fin faire****Fin pour** ( $i \in 1, 2$ )**Fin faire****Si**  $p_1$  est associe a des primitives placees plus bas dans la hierarchie**Faire** $l_1 \leftarrow$  l'ensemble de ces primitives $l_2 \leftarrow$  l'ensemble des primitives associees a  $p_2$ 

qui lui sont plus bas dans la hierarchie

correspondance-globale( $l_1, l_2, l'$ )ajouter  $l'$  a  $l$ **Fin faire****end proc**

Figure 5.7: Algorithme de mise en correspondance de primitives

**Les éléments incertains** qui constituent l'information moins fiable du système.

Ces éléments peuvent être vus comme des interprétations partielles possibles de la scène, au vu des données acquises par les capteurs.

**Les observations (ou informations sensorielles)** qui regroupent l'ensemble des données récentes acquises par les capteurs. Le système doit intégrer l'information véhiculée par ces données.

Les informations sensorielles parviennent au système sous forme d'un flux de données asynchrone. Le processus d'intégration fonctionne de façon cyclique. Chaque cycle d'intégration est précédé d'une phase d'attente, pendant laquelle le système attend de nouvelles observations. L'arrivée de nouvelles données déclenche l'exécution d'un nouveau cycle d'intégration.

Au cours d'un cycle, le processus d'intégration sensoriel exploite les observations pour réaliser deux buts:

1. l'affinement de l'information sûre, par mise à jour des paramètres numériques des éléments sûrs, et amélioration de leur précision.
2. la confirmation ou l'infirmerie de l'information incertaine, et par conséquent l'enrichissement de l'information sûre, par addition de nouveaux éléments.

Nous décrivons dans le paragraphe qui suit, l'historique d'un cycle d'intégration.

### 5.5.2 historique d'un cycle d'intégration

Pour réaliser ces deux buts, l'algorithme d'intégration procède en 5 étapes:

1. mise en correspondance des observations avec les éléments sûrs. Le système essaie ainsi d'interpréter ses dernières observations avec le modèle.
2. mise en correspondance des éléments incertains avec les observations. Le système essaie de confronter ses hypothèses (ou interprétations) courantes, afin de les consolider.
3. génération d'hypothèses à partir d'observations. Le système considère les observations inutilisées comme une information nouvelle, indépendante de ce qu'il a enregistré jusqu'ici.



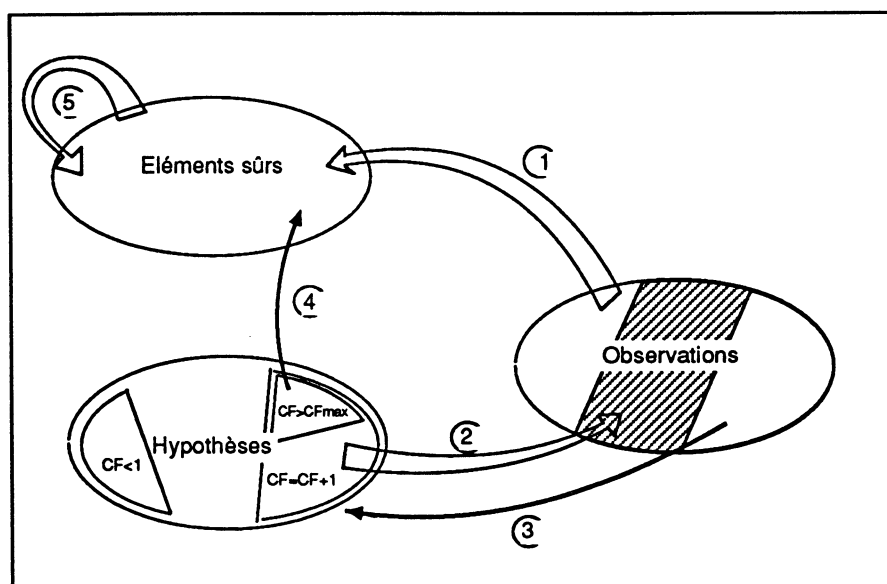


Figure 5.8: Processus cyclique d'intégration

4. validation des hypothèses. Le système met à jour sa base d'hypothèses en oubliant les hypothèses trop peu fiables, et en transformant en éléments sûrs les hypothèses suffisamment fiables.
5. mise en correspondance du modèle avec lui-même. Le système "recolle" les segments 3D et les éléments de surfaces, que les artefacts des techniques d'inférence de forme ont morcelés.

Ces différentes étapes se traduisent par des échanges d'informations entre les trois ensembles définis au paragraphe 5.5.1. Nous avons représenté ces flux par des flèches dans la figure 5.8. Nous détaillons dans la section qui suit les différentes opérations mises en œuvre, lors de chacune des étapes du cycle d'intégration.

(Notation: Les éléments sûrs sont notés  $s$ . Les éléments incertains sont notés  $i$ . Les observations sont notées  $o$ .)

1. **Mise en correspondance des observations avec les éléments sûrs:**  
pour toute primitive géométrique observée  $o$ , on recherche les éléments sûrs  $s$  qui lui correspondent. Lorsqu'il existe une correspondance unique  $(o, s)$ , les deux primitives  $o$  et  $s$  sont fusionnées, la valeur de  $s$  est remplacée par la valeur obtenue par fusion. L'observation  $o$  est détruite. Lorsqu'il existe

au moins deux correspondances, L'observation  $o$  est détruite. Lorsqu'il n'y a aucune correspondance, on passe à l'observation suivante.

2. **Mise en correspondance des éléments incertains avec les observations:** pour chaque primitive géométrique incertaine  $i$ , on recherche les éléments observés  $o$  qui lui correspondent. Pour chaque correspondance  $(i, o)$  trouvée, le système construit une primitive géométrique  $i'$ , par fusion des deux primitives  $i$  et  $o$ , génère une hypothèse qui permet de désigner cette primitive, et lui associe un facteur de confiance  $CF_{i'} = CF_i + F_{\text{capteur}}$ , où  $F_{\text{capteur}}$  est un nombre qui quantifie la fiabilité du capteur. Lorsque  $i$  n'admet aucune correspondance, son facteur de confiance est diminué de 1.
3. **Génération d'hypothèses à partir d'observations:** les observations qui ont survécu aux étapes précédentes, sont ajoutées à l'ensemble des éléments incertains avec le facteur de confiance  $CF_o = F_{\text{capteur}}$
4. **Validation des hypothèses:** pour chaque primitive géométrique incertaine  $i$ , on examine le facteur de confiance  $CF_i$  de l'hypothèse qui la désigne. Si celui-ci est supérieur à  $CF_{\text{max}}$ <sup>2</sup>, et qu'il n'est pas ambigu<sup>3</sup>, on le valide. Nous pouvons alors mettre à jour les valeurs des hypothèses mères. En effet,  $i$  a été construite en fusionnant les valeurs  $i_1$  et  $i_2$  de ses hypothèses mères. Les valeurs  $i, i_1$  et  $i_2$  constituant des estimateurs différents d'une même entité géométrique, il est possible de remplacer chacune de leur occurrence dans le modèle, par le meilleur d'entre eux, à savoir  $i$ . Nous avons vu dans le chapitre 4 que ces modifications, se propagent par le biais des relations de dépendance, et induisent d'autres modifications dans le graphe des hypothèses. Si par exemple,  $i_1$  et  $i_2$  sont deux segments 3D, lorsqu'ils sont remplacés par le segment 3D  $i$ , le système de gestion des hypothèses va vérifier la validité des relations de parallélisme et d'appartenance que  $i_1$  entretenait avec les éléments de surface. Les relations qui ne sont plus valides sont détruites. Une fois ces modifications réalisées,  $i$  est retirée de l'ensemble des éléments incertains

<sup>2</sup> $CF_{\text{min}}$  est défini comme le seuil du facteur de confiance en-dessous duquel l'hypothèse est rejetée par le système, et  $CF_{\text{max}}$  le seuil au-dessus duquel l'hypothèse est considérée comme vraie. (voir paragraphe 4.3.2)

<sup>3</sup>une hypothèse est ambiguë s'il existe d'autres hypothèses qui lui sont incompatibles.

et rajoutée à l'ensemble des éléments sûrs. Si  $CF_i$  est inférieur à  $CF_{min}$ , on détruit  $i$ .

5. **Mise en correspondance du modèle avec lui-même:** pour toute primitive géométrique sûre  $s$ , on cherche ses correspondants sûrs eux-aussi. Lorsque  $s$  n'admet qu'un seul correspondant  $s'$ , on fusionne  $s$  et  $s'$ . L'élément  $s''$  obtenu hérite des relations géométriques de  $s$  et  $s'$ . Les primitives  $s$  et  $s'$  sont alors détruites.

## 5.6 Résumé

Nous avons introduit une batterie de procédures, qui permettent de confronter deux primitives géométriques. L'application de ces procédures à deux primitives, indiquent la possibilité que ces deux primitives soient liées par une relation géométrique, ou que ces deux primitives constituent deux estimateurs différents d'une même entité physique. Ces procédures permettent d'effectuer une **mise en correspondance locale**. La **mise en correspondance globale** consiste à rechercher toutes les mises en correspondance locales  $(e_i, f_i)$  entre deux ensembles  $E = \{e_i \mid i \in \{1, \dots, n\}\}$  et  $F = \{f_j \mid j \in \{1, \dots, m\}\}$  de primitives.

Nous avons présenté une stratégie de contrôle, permettant de guider la recherche des mises en correspondance locales et d'éviter une recherche exhaustive. Cette stratégie de contrôle tient compte de la structure hiérarchique induite par les relations géométriques.

Nous avons ensuite présenté le processus d'intégration comme un processus cyclique, au cours duquel s'opèrent des échanges entre trois bases d'informations: l'ensemble des éléments sûrs, l'ensemble des éléments incertains, et l'ensemble des observations. Les échanges par une séquence de mises en correspondance globales et de transferts de primitives géométriques entre les différentes bases.

A chaque cycle d'intégration, la base des éléments sûrs se s'affine, et s'enrichit. La base des éléments incertains est mise à jour.

## Chapitre 6

# Application: la caméra mobile

Dans ce chapitre, nous décrivons la mise en œuvre du système de perception dans un premier cas pratique: la caméra mobile.

Les deux applications sont abordées sous une approche multisensorielle. Dans le cas de la caméra mobile, la scène est statique. Les différentes vues acquises par la caméra au cours de son déplacement, sont considérées comme des sources indépendantes d'informations. Le système que nous avons développé, construit et met à jour un modèle tri-dimensionnel de la scène en intégrant ces informations.

Nous présentons d'abord le principe de l'expérience, puis décrivons l'architecture matérielle mis en œuvre. Nous introduisons ensuite une technique d'inférence de segments 3D, à partir de vues monoculaires. Ces segments 3D vont constituer des observations que le système d'intégration utilisera, pour construire et mettre à jour le modèle composite de la scène. Nous présentons enfin les résultats obtenus.

## 6.1 Principe

Le système que nous avons développé, construit une description tri-dimensionnelle de la scène du type "fil de fer", en exploitant une séquence d'images. Ces images proviennent d'une caméra, fixée sur la pince d'un robot manipulateur.

Nous supposons que la scène est statique. Par contre, aucune hypothèse n'est émise quant à l'amplitude de déplacement de la caméra. La figure 6.1 représente le montage robot-caméra.

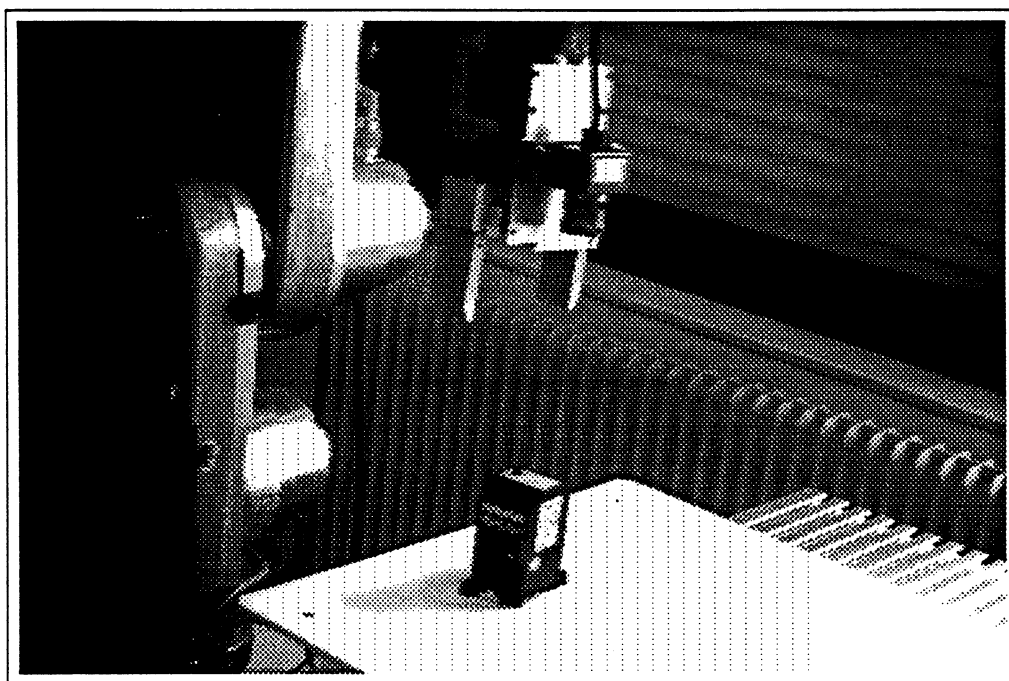


Figure 6.1: Montage caméra-robot manipulateur

## 6.2 Architecture matérielle

La partie matérielle du système est constituée d'un robot manipulateur à 6 axes SCEMI, programmable en langage LM à partir d'un ordinateur HP1000.

Une caméra PULNIX CCD est fixée sur la pince du robot.

Le signal vidéo produit par la caméra, est numérisé par une carte fg150 d'IMAGING TECHNOLOGY. Cette carte est connectée à une station de travail SUN3/160.

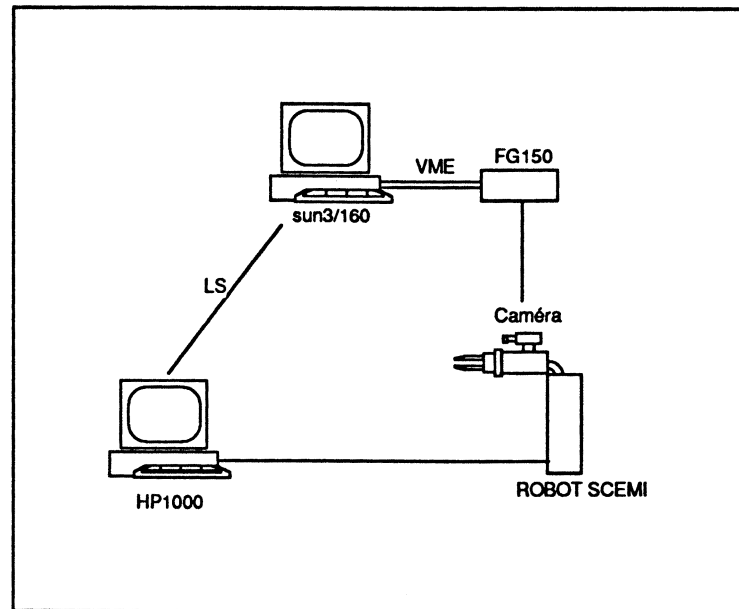


Figure 6.2: Architecture matérielle

L'architecture matérielle est schématisée figure 6.2.

La communication entre l'ordinateur HP1000 et la station SUN est réalisée via une ligne série RS-232, et permet l'échange de données entre les deux stations, ainsi que la commande à distance du robot à partir de la station SUN.

Dans la section suivante, nous présentons une technique d'inférence de segments 3D, à partir d'une vue monoculaire.

### 6.3 Obtention des données tri-dimensionnelles

Rappelons que pour mettre à jour le modèle composite, le module d'intégration impose que les observations à intégrer, soient décrites à l'aide des primitives définies dans la section 3.2. Nous nous proposons de décrire les observations à l'aide de segments 3D.

Pour cela, trois étapes sont nécessaires:

1. le calibrage du système.
2. l'extraction de segments 2D à partir des images.

3. la projection inverse des segments 2D, pour inférer des segments 3D.

Ces étapes font l'objet des paragraphes suivants.

### 6.3.1 Calibrage du système

Le calibrage d'un capteur est une opération cruciale pour la bonne exécution des processus, qui en amont, sont chargés d'exploiter les informations fournies.

Le calibrage permet de modéliser analytiquement la relation qui existe entre les mesures acquises par le capteur, et des entités physiques qui influencent ces mesures.

#### Calibrage du robot

Le bras manipulateur déplace son extrémité terminale, en articulant ses éléments autour de ses axes de rotation. Chacun de ces axes est muni d'un codeur. Le contrôleur de bras traite les signaux produits par les codeurs, pour fournir la position et l'orientation du repère de la pince (appelé "repère robot"), par rapport à un repère fixe (appelé "repère station"). Ce dernier est généralement situé dans le socle du robot. Ces informations sont accessibles à l'utilisateur sous forme d'une matrice homogène. Le contrôleur de robot met à jour cette matrice, en utilisant les signaux délivrés par les codeurs, après chaque mouvement du bras.

La principale source d'erreur sur le calcul de la matrice homogène, provient de l'atteinte des positions en butées, des articulations du robot.

Pour cette raison, on effectue un recalage du robot, au début de chaque expérience. Cette opération consiste à placer le bras dans une position particulière, où toutes les articulations sont en butée. Dans cette position, la matrice homogène est connue à l'avance avec précision. La position de la pince fournie par le contrôleur est alors initialisée à l'aide de cette matrice.

De plus, au cours de l'expérience, on doit s'assurer que les articulations du robot ne parviennent en aucun cas à leurs positions de butée.

### Calibrage de la caméra

Le calibrage de la caméra a pour but de calculer ses paramètres. On distingue deux types de paramètre:

**les paramètres intrinsèques:** la distance focale, les facteurs d'échelle (qui permettent de retrouver la taille réelle des pixels), la position du centre de l'image.

**les paramètres extrinsèques:** la rotation et la translation du déplacement rigide, qui permet de passer du repère de la caméra au repère de la scène.

Pour calculer ces paramètres, nous utilisons une technique présentée dans l'article [FT86]. Nous en rappelons ici simplement le principe.

Dans un premier temps, les auteurs analysent les équations reliant les coordonnées des points de la scène, aux coordonnées de leur point image. Cette analyse montre que dans l'espace des coordonnées homogènes, ces équations, appelées "équations de transformation perspective", sont linéaires. L'analyse permet également d'établir les relations qui existent entre les paramètres de la caméra, et les coefficients de la matrice de transformation perspective. Ainsi, il est possible de retrouver les valeurs des paramètres cherchés, à partir de la matrice de transformation perspective.

Dans un deuxième temps, les coefficients de la matrice sont calculés expérimentalement, à partir d'un ensemble de points de la scène, dont on connaît précisément les coordonnées scène (points situés sur une mire) et les coordonnées image (détection automatique des points de la mire sur l'image). Pour cela, les auteurs écrivent les équations linéaires pour chacun des points de la mire:

$$U_i = M'X_i \quad (6.58)$$

(où  $X_i$  sont les coordonnées du  $i$ ème point de la mire) Celles-ci forment un système d'équations sur-déterminé, où les inconnues sont les coefficients de  $M'$ .

La calibration est alors réalisé en résolvant l'équation 6.58, puis en utilisant les coefficients de la matrice  $M'$ , pour calculer les paramètres intrinsèques et extrinsèques.

Une description plus détaillée de la méthode de calibrage est donnée dans l'annexe A.



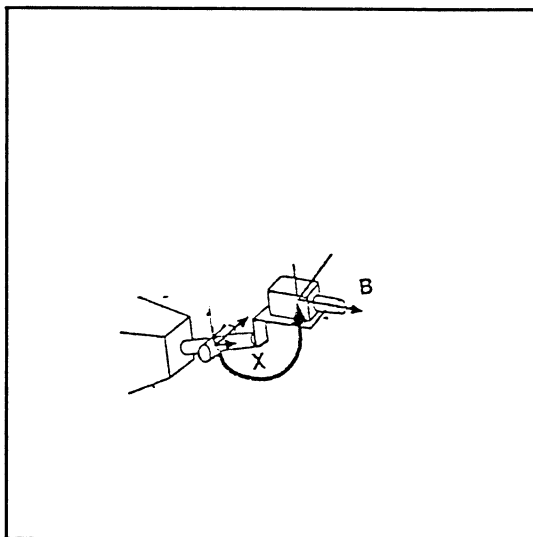


Figure 6.3: Calcul du lien caméra-robot

### Calibrage du lien robot-caméra

Ce calibrage permet de calculer la matrice de passage  $X$  (figure 6.3), permettant de passer du repère de la pince du robot, au repère de la caméra.

La technique que nous utilisons, est développée dans l'article [SA86]. Rappelons en simplement le principe:

On effectue un calibrage simple de caméra dans trois positions différentes du bras. Le calcul des paramètres extrinsèques, et la connaissance des trois positions de la pince, permet de former un système d'équations dont la matrice  $X$  est solution.

La méthode de calibration est décrite plus en détail dans l'annexe B.

#### 6.3.2 Traitement des images

Le traitement de l'image permet d'extraire de l'image brute, considérée comme une matrice de pixels, l'information pertinente sous forme d'un ensemble de segments de droite.

Ce traitement s'effectue en trois étapes que nous décrivons brièvement dans les paragraphes suivants:

**détection des points de contraste**

Les points de contraste sont détectés à l'aide du filtre récursif de Deriche ([Der87]). Ce filtre est optimal au regard de trois critères:

1. il maximise le rapport signal/bruit.
2. il localise au mieux le point de contraste.
3. il limite le nombre de pics.

Ces critères sont combinés pour en former un unique, appelé critère de Canny [Can83]. Deriche montre que la solution optimale, peut-être approchée par un filtre récursif.

**chaînage des points de contraste**

Le processus de chaînage parcourt l'image, ligne par ligne en une passe, et permet de regrouper les points de contraste en lignes de contraste.

**extraction des segments 2D**

Chaque ligne est approximée par un ensemble de segments de droite.

Les segments de droite obtenus par la séquence des opérations que nous venons de décrire, approximent donc les lieux de l'image où l'intensité lumineuse varie fortement dans une direction. Ces changements d'intensité lumineuse dans les images, sont attribués à des discontinuités de profondeurs, des changements d'orientation, de réflectance, d'illumination ou de couleur, dans la scène. Chaque segment 2D de l'image correspond donc dans la scène, à un segment 3D approximant une arête d'un objet, une ligne de discontinuité de surface, ou une ligne de changement de luminance.

Différents travaux supportent l'importance de la sensibilité aux alignements dans le processus visuel humain, en avançant une argumentation d'ordre physiologique. On pourra se reporter aux articles [Mar82], [Zuc85], [HW77]

### 6.3.3 Technique de la projection inverse

Dans la section 3, nous avons insisté sur le fait que les données fournies par les capteurs, devaient impérativement s'exprimer au moyen des primitives géométriques introduites dans la section 3. Précédemment, nous avons expliqué la manière dont on pouvait connaître la position courante de la caméra dans la scène, à partir des informations émises par le contrôleur du robot (6.3.1), puis nous avons vu que l'image brute pouvait être traitée, afin d'en extraire un ensemble de segments 2D.

Nous présentons maintenant, une technique permettant d'inférer des segments 3D, à partir des segments 2D extraits et de la position de la caméra.

Nous esquissons les principes de cette technique dans ses grandes lignes:

#### Principe

L'opération de projection inverse consiste à calculer la position d'un point dans l'espace tri-dimensionnel, lorsqu'on connaît la position de son image dans le plan rétinien.

L'opération de projection perspective n'est pas bijective: plusieurs points dans l'espace peuvent se projeter sur le même point dans l'image. Tous ces points sont situés sur la droite, qui passe par le centre focal et le point image. En pratique, la projection inverse doit donc associer à chaque point de l'image, une droite et non un point unique dans l'espace 3D. Nous pouvons faire deux remarques:

- Dans le contexte particulier de notre expérience, la profondeur des points visibles est bornée. En effet, les objets vus par la caméra sont posés sur la table de travail. La profondeur minimale est donnée par la distance entre le centre focal et la lentille extérieure de l'objectif. La profondeur maximale est déterminée par la distance entre le centre focal et le point de la table le plus éloigné de la caméra. Par conséquent, les points 3D correspondant à un point image, ne constituent plus une droite, mais un segment de droite.
- Le concept de variable aléatoire multi-dimensionnel que nous avons introduit dans le chapitre 2, permet de représenter un segment de droite, par un point

aléatoire. Ce point aléatoire est défini par une distribution dont tous les points du segment constituent les occurrences possibles. En toute rigueur, la distribution réelle du point aléatoire est uniforme (toutes les occurrences ont la même densité de probabilité). Néanmoins, il est possible d'approximer cette distribution uniforme, à l'aide d'une distribution normale (ou gaussienne). Nous avons vu au chapitre 2, que cette modélisation avait l'avantage d'être compacte et aisée à manipuler.

Pour effectuer la projection inverse d'un point de l'image, nous construisons un point aléatoire de l'espace 3D, en utilisant les équations de la projection perspective. Celui-ci est obtenu dans le repère de la caméra. Les informations fournies par le contrôleur du robot, permettent d'effectuer un changement de repère, afin de situer ce point dans le repère de la scène.

L'opération de projection inverse est appliquée aux deux extrémités d'un segment 2D de l'image, pour inférer le segment 3D de la scène lui correspondant.

Pour plus de détails, on pourra se reporter à l'annexe C.

#### **6.3.4 Expérimentation**

Nous avons appliqué la technique de projection inverse sur une vue simple d'un objet posé sur une table (voir figure 6.4). Le résultat des étapes successives du traitement, est visualisé sur les figures suivantes:

##### **Image en niveaux de gris**

L'image est digitalisée en  $512 \times 512$  pixels, et numérisée sur 256 niveaux de gris. (voir figure 6.5)

##### **Extraction des contours**

Les points de contraste sont extraits à l'aide du filtre de Deriche (voir figure 6.6), puis chaînés pour former les contours (ou lignes de contraste)

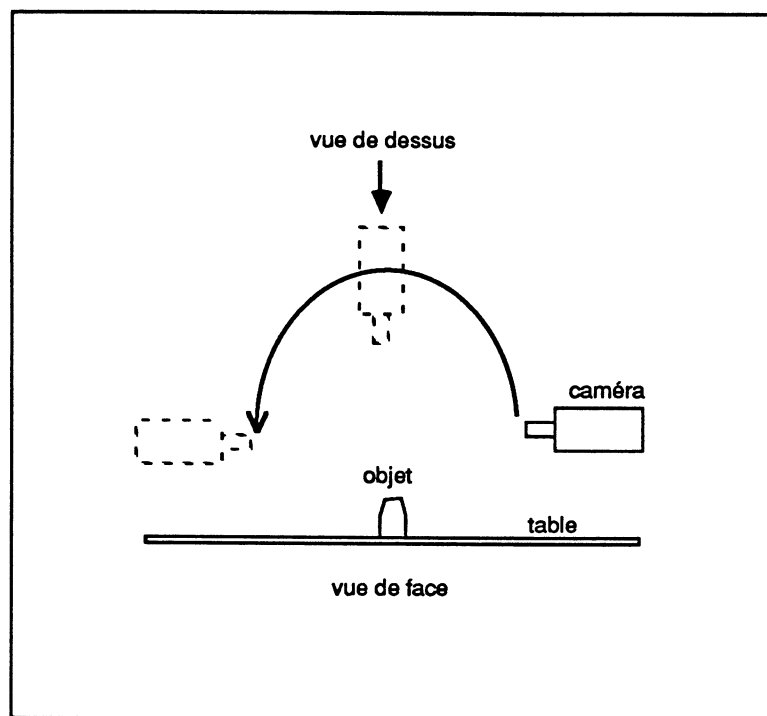


Figure 6.4: Configuration de l'expérience

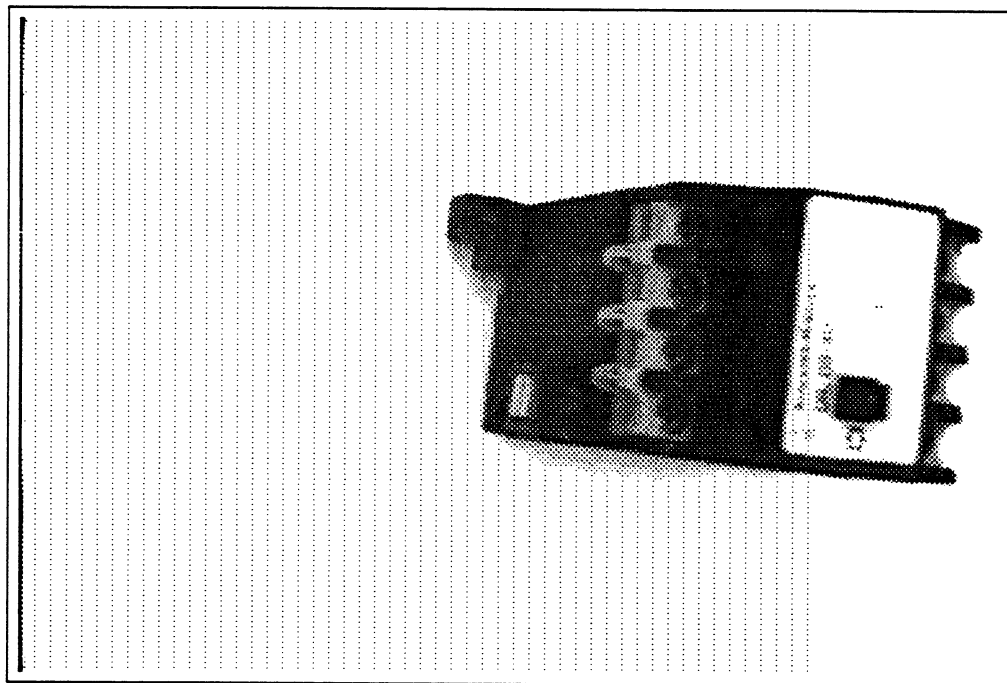


Figure 6.5: Image en niveau de gris

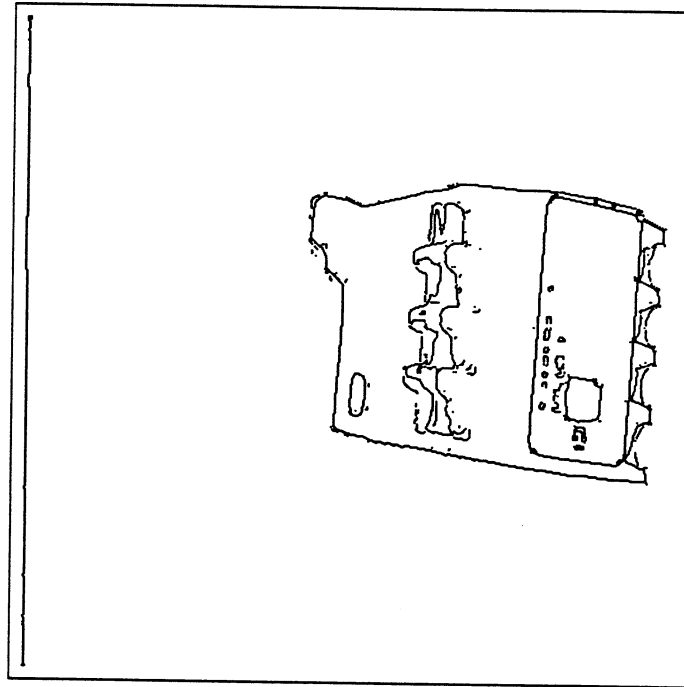


Figure 6.6: Extraction des contours

### Extraction de segments 2D

Les contours sont approximés par des lignes polygonales, pour former les segments 2D (voir figure 6.7)

### Projection inverse des segments 2D

Nous avons visualisé la vue de dessus des segments 3D obtenus, en faisant apparaître l'ellipsoïde d'imprécision du point milieu du segment 3D (voir figure 6.8). Les directions principales des ellipsoïdes, qui matérialisent la distribution de position du point milieu des segments 3D, convergent vers le centre focal. C'est en effet, une conséquence de la géométrie de projection perspective.

Nous avons grossi un segment isolé, pour mettre en évidence la forme ellipsoïdale de l'imprécision (voir figure 6.9).

La figure 6.10 représente une vue de côté des segments 3D obtenus. On peut remarquer que les segments sont alignés sur un plan. Ce dernier n'est autre que le plan perpendiculaire à la direction focale, situé à une distance  $Z_0$  par défaut du centre

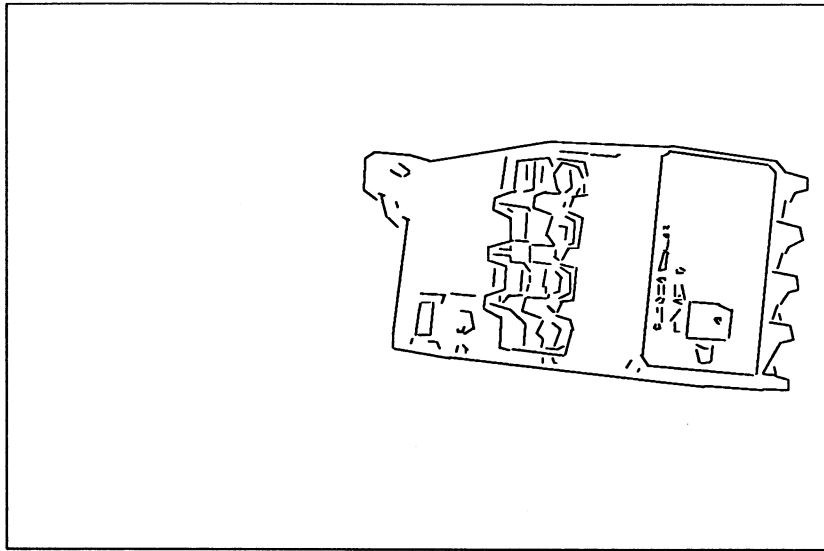


Figure 6.7: Extraction de segments 2D

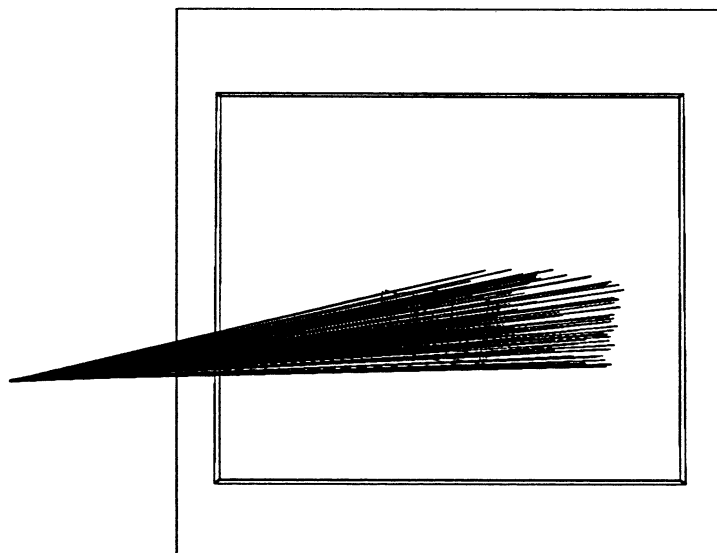


Figure 6.8: Projection inverse des segments images. Vue de dessus

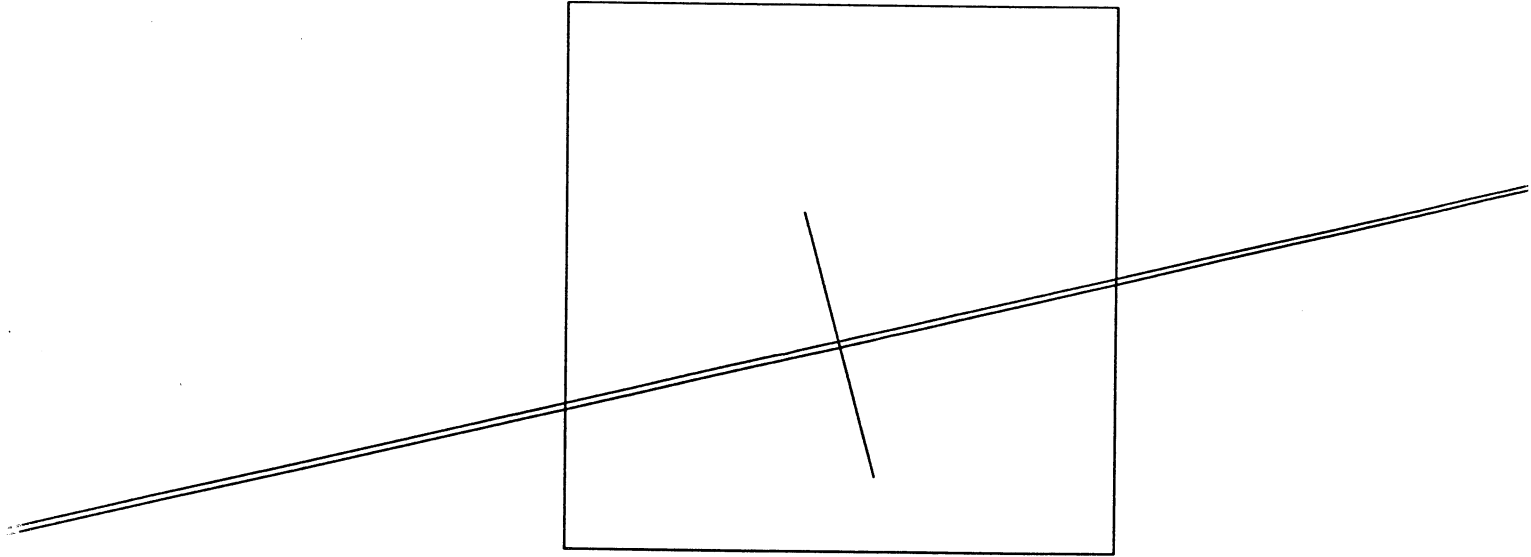


Figure 6.9: Vue d'un segment 3D et de l'imprécision du point milieu

focal.

## 6.4 Résultats expérimentaux

Dans les paragraphes suivants, nous discutons de l'applicabilité à notre problème, des opérations d'intégration définies dans le chapitre 5, ensuite nous décrivons l'état initial du système puis son évolution au cours des cycles d'intégration.

### 6.4.1 Remarque préliminaire

Les conditions nécessaires nous permettant d'utiliser les techniques introduites au chapitre 5 sont satisfaites, exceptée la condition d'indépendance des précisions des paramètres, qui est requise lors de la fusion de deux segments 3D.

En effet, l'opération de fusion de deux segments 3D n'est réalisable qu'à la condition que l'imprécision de leurs extrémités soient indépendantes. Pour étudier cette indépendance, nous sommes amenés à rechercher l'origine de ces imprécisions. On peut citer trois sources distinctes:

1. celle due à la discrétisation de l'image. Cette dernière associe à un point image, une imprécision de l'ordre de la taille d'un pixel.



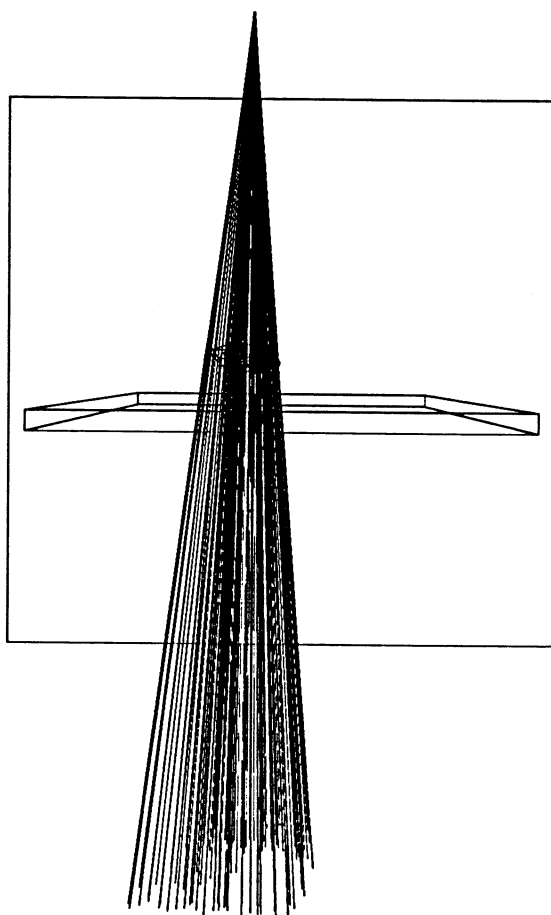


Figure 6.10: Projection inverse des segments images, vue de côté

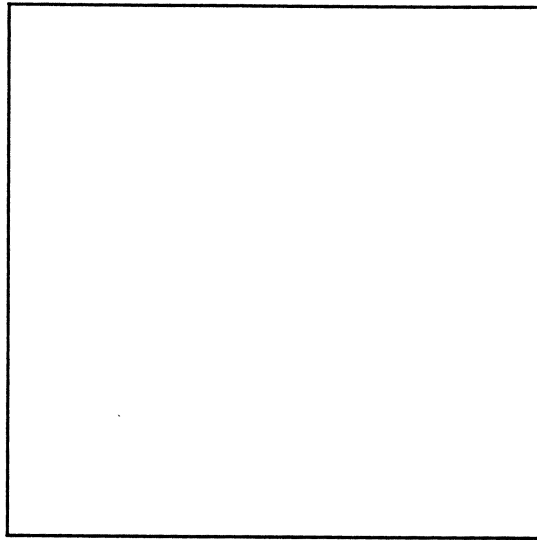


Figure 6.11: Etat initial du système

2. celle due à l'imprécision de calibrage du système.
3. celle due à l'imprécision des informations fournies par le contrôleur de bras, sur la position de la pince.

Les deuxième et troisième composantes de l'imprécision sont identiques pour tous les points 3D obtenus par projection inverse. Ainsi, ceux-ci induisent une dépendance entre l'imprécision de deux extrémités d'un même segment 3D.

Cependant, nous allons supposer que l'imprécision de calibrage ainsi que celle due au contrôleur sont négligeables devant celle liée à la discrétisation de l'image. Une étude visant à analyser l'importance de ces imprécisions devrait le confirmer par la suite.

Sous couvert de cette hypothèse, les imprécisions des extrémités des segments 3D sont indépendantes, et les techniques développées au chapitre 5 sont bien applicables.

#### 6.4.2 Initialisation et amorçage du processus d'intégration

Nous avons représenté figure 6.11, l'état initial du système. L'ensemble des éléments sûrs et l'ensemble des hypothèses sont vides.

### Premier cycle d'intégration

Lors du premier cycle d'intégration, l'ensemble des observations est constitué de la liste des segments 3D obtenus par projection inverse de la première image. Les premières étapes du premier cycle d'intégration, présentées au paragraphe 5.5.2, n'ont aucun effet sur les données, puisque les ensembles sont vides. L'exécution du premier cycle transforme uniquement les premières observations, en hypothèses.

### Deuxième cycle d'intégration

Lors du deuxième cycle d'intégration, l'ensemble des hypothèses contient les segments 3D provenant de la première vue. L'ensemble des observations contient les segments 3D provenant de la seconde. L'ensemble des éléments sûrs étant toujours vide, l'étape réellement effective du cycle d'intégration, est donc celle de mise en correspondance des hypothèses et des observations.

Rappelons que pour chaque couple  $(h, o)$  en correspondance, le système génère une hypothèse dont la valeur est le segment 3D obtenu en fusionnant les segments 3D désignés par  $h$  et  $o$  (le test de mise en correspondance de deux segments 3D est détaillé dans le paragraphe 5.2.1. La procédure de fusion est décrite dans le paragraphe 5.3.2). Le facteur de confiance de l'hypothèse générée est obtenu en incrémentant d'une unité, le facteur de confiance de  $h$ .

Les segments 3D générés lors de cette étape sont visualisés dans la figure 6.12.

On peut remarquer que la plupart des segments 3D générés ne sont pas réels. En effet, le système cherche toutes les correspondances possibles. Parmi elles, certaines sont exactes (l'hypothèse  $h$  et l'observation  $o$  relèvent du même segment physique), mais certaines sont erronées. Un segment 3D obtenu par fusion de  $h$  et  $o$ , où  $(h, o)$  est une correspondance erronée, est donc fictif.

Rappelons que le système tient compte de la nature incertaine de ces segments en les enregistrant par le biais d'une hypothèse. Cette hypothèse sera confirmée ou infirmée par les observations acquises aux cycles suivants.

De plus, le système considère incompatibles, les correspondances incluant une même hypothèse. Ce mécanisme est à rapprocher des ambiguïtés d'interprétation d'une

vue monoculaire. Les hypothèses générées par deux correspondances incompatibles sont déclarées incompatibles. La figure 6.13 illustre une configuration où deux correspondances sont incompatibles. L'incompatibilité est enregistrée à l'aide des relations définies au paragraphe 4.3.3.

### cycles d'intégration suivants

A chaque cycle d'intégration, les hypothèses générées par la mise en correspondance des hypothèses et des observations, possèdent un facteur de confiance de plus en plus élevé. Tant que ce facteur de confiance n'atteint pas la valeur  $CF_{max}$ , le système se contente de confirmer et infirmer les hypothèses existantes, et de transformer les observations non utilisées, en hypothèses. (Rappelons que  $CF_{max}$  définit le seuil de confiance à partir duquel l'hypothèse est validée.)

Dans notre implémentation,  $CF_{max} = 6$ . Il nous faut donc attendre le sixième cycle, pour que des hypothèses soient susceptibles d'être validées. Pour qu'une hypothèse soit validée, il est nécessaire que sa valeur de confiance soit supérieure à  $CF_{max}$ , et qu'il n'existe pas d'hypothèse possédant un facteur de confiance aussi élevé, qui lui soit incompatible.

#### 6.4.3 Evolution du modèle

Les premiers éléments que le système considère comme étant sûrs, apparaissent au sixième cycle d'intégration. Nous les visualisons sur la figure 6.14. Les droites qui limitent la surface de la table ont été tracées à la main, afin de montrer que le système de perception localise bien les segments 3D, dans un repère absolu indépendant des positions successives de la caméra.

L'ensemble des éléments sûrs du modèle à l'issue du huitième cycle d'intégration est représenté dans la figure 6.15.

L'ensemble des éléments sûrs du modèle à l'issue du dixième cycle d'intégration est représenté dans la figure 6.16.

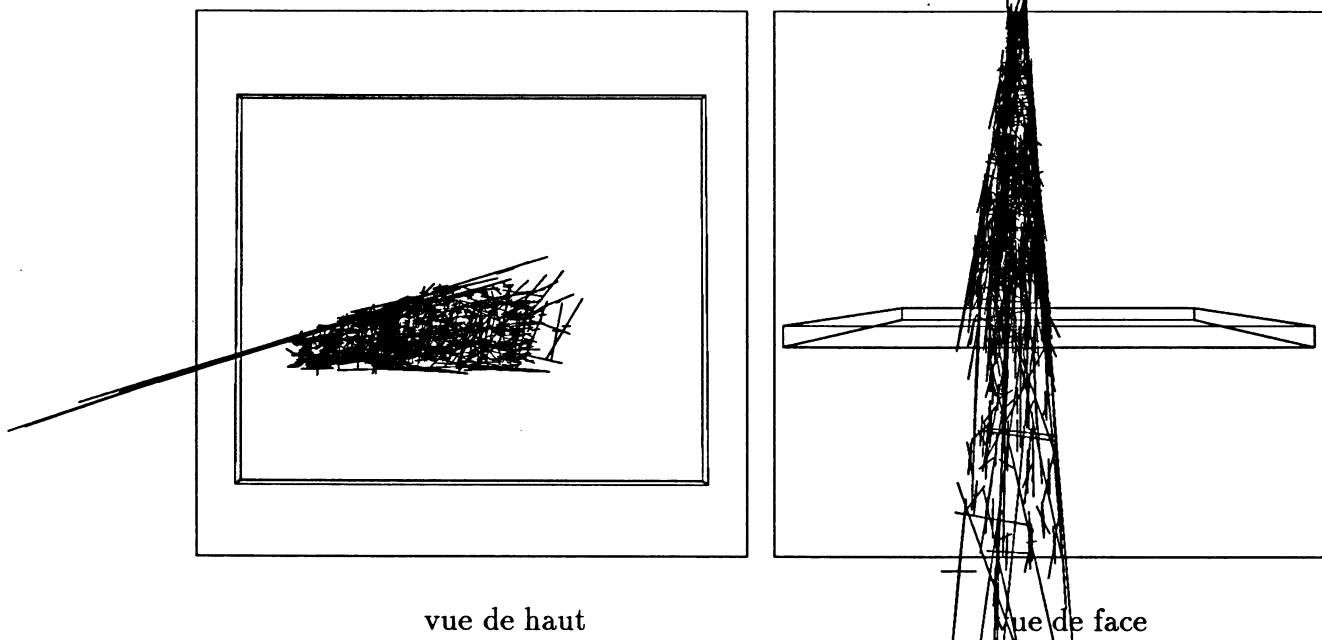


Figure 6.12: segments 3D générés lors du second cycle

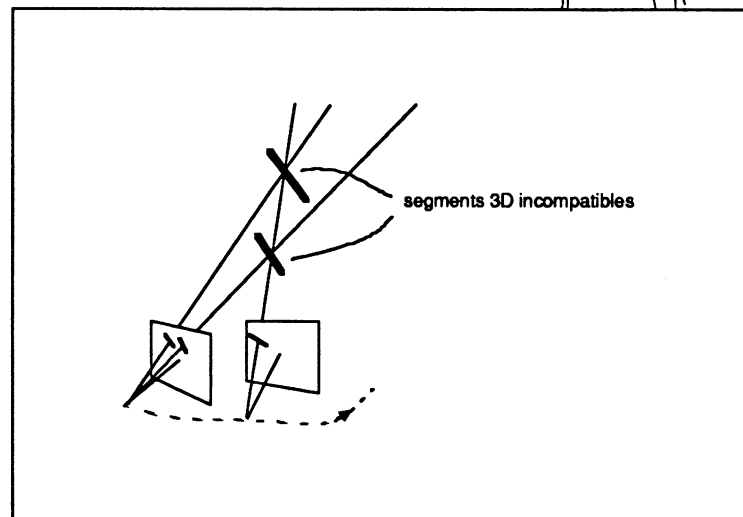


Figure 6.13: Correspondances incompatibles entre segments 3D

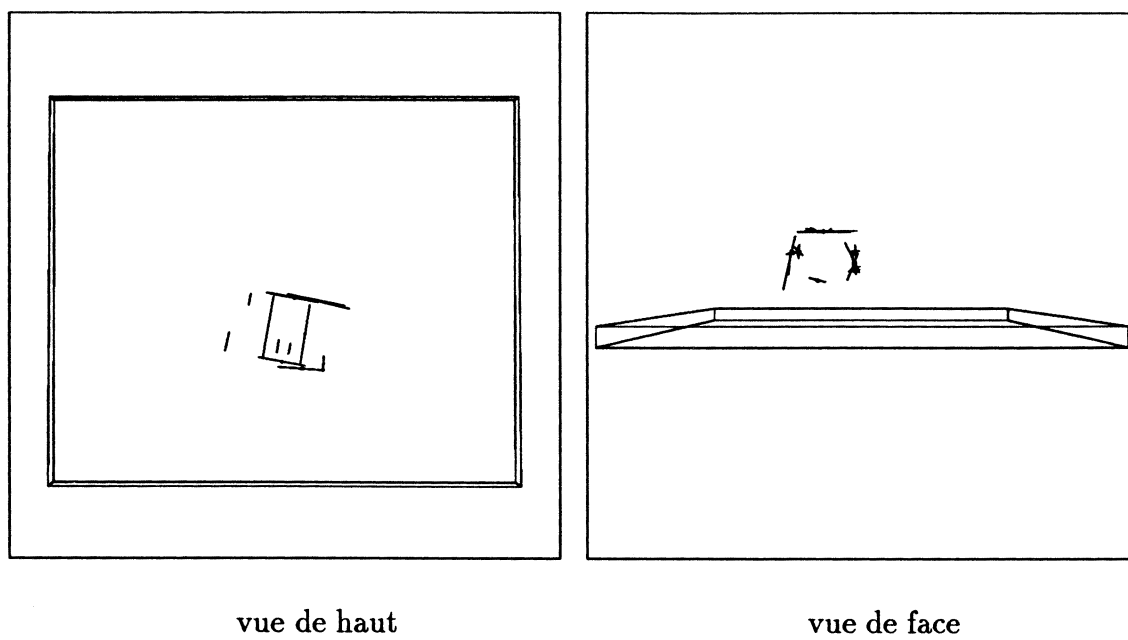


Figure 6.14: Premiers segments 3D sûrs

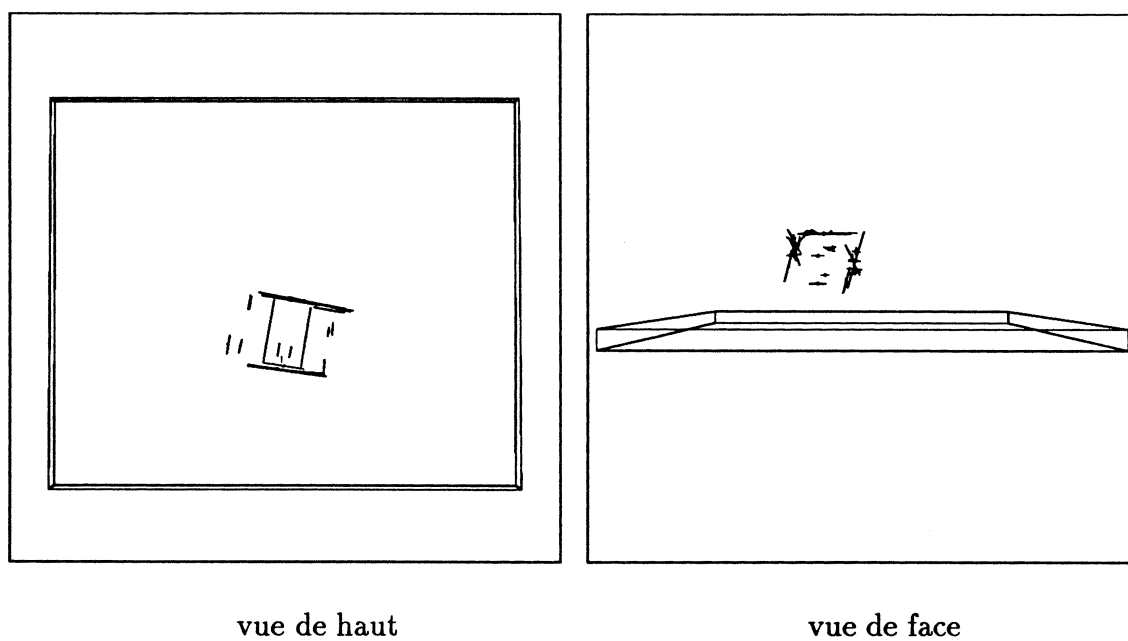


Figure 6.15: Eléments sûrs après 8 cycles d'intégration

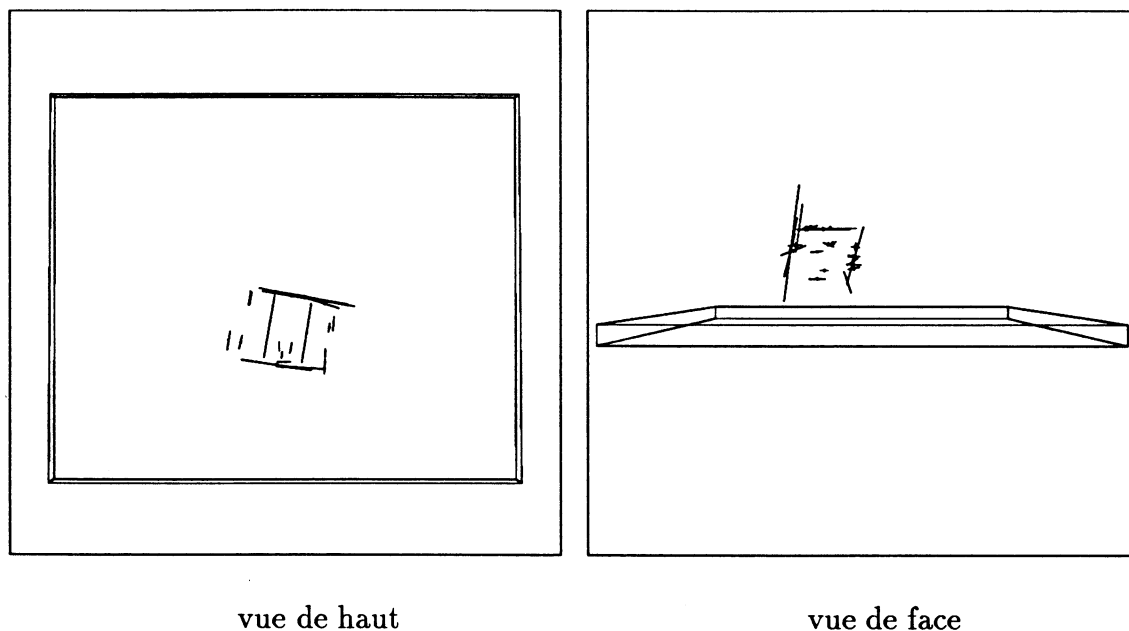


Figure 6.16: Eléments sûrs après 10 cycles d'intégration

#### 6.4.4 Autres résultats

Les résultats suivants concernent la reconstruction tri-dimensionnelle d'un "rubik's cube" (voir figure 6.17). La figure 6.18 présente les segments 2D obtenus par extraction de contours, puis approximation polygonale.

Les premiers segments 3D sûrs apparaissent au sixième cycle d'intégration, et sont visualisés figure 6.19. La figure 6.20 visualise l'ensemble des segments 3D sûrs après huit cycles d'intégration. La figure 6.21 visualise l'ensemble des segments 3D sûrs après dix cycles d'intégration.

Le nombre de segments 2D dans chaque image et de segments 3D dans le modèle pour chacun des cycles d'intégration sont indiqués figure 6.22

## 6.5 Discussion

Lors de ces expériences, nous avons ignoré l'imprécision due aux mouvements du bras manipulateur. Lorsque l'opérateur commande au robot de se placer en une position particulière, ce dernier s'exécute et renvoie par l'intermédiaire du contrôleur de bras, la position atteinte. Il existe toujours un écart entre la position commandée et la position mesurée par le contrôleur. Aussi, la position de la pince qui est effectivement prise en compte dans le calcul des segments 3D, lors de l'opération de projection inverse, est celle fournie par le contrôleur, et non celle qui est commandée. Néanmoins, il existe toujours une erreur résiduelle entre la position réellement atteinte par le bras, et la position mesurée par le contrôleur de bras. Celle-ci est essentiellement due à l'erreur d'échantillonnage provenant des différents codeurs. Cependant, ces erreurs demeurent faibles tant que le nombre de mouvements cumulés du bras reste faible. Dans ces conditions, l'imprécision liée aux erreurs de position du bras est négligeable devant celle liée aux erreurs de calibrage de la caméra. C'est la raison pour laquelle, nous nous sommes limités à une dizaine de mouvements élémentaires du bras.

Nos réalisations pratiques nous ont permis de valider partiellement notre approche. En particulier nous avons prouvé les points suivants:

1. le cadre que nous proposons est assez souple pour accepter des données très imprécises. En effet, les informations tri-dimensionnelles produites par une analyse monoculaire sont incomplètes, et ne fournissent aucune donnée quantitative sur la profondeur.
2. les ambiguïtés de correspondance entre les hypothèses courantes et les nouvelles observations, étaient importantes et typiques des processus d'interprétation de vues monoculaires. L'importance de ces ambiguïtés a fourni au module de description et de gestion des incertitudes, un contexte d'expérimentation particulièrement sévère.

Malheureusement, faute de données disponibles, nous n'avons pas pu tester la capacité de notre système à accueillir des informations de surface, telles que celles produites par des systèmes à ultra-son, ou des systèmes utilisant un faisceau laser.



Néanmoins, l'intégration d'éléments de surface et la structuration qu'elle induit au niveau du modèle géométrique, devrait faciliter l'intégration des segments 3D (voir paragraphe 5.4)

## 6.6 Résumé

Nous avons expérimenté notre approche dans une situation réelle.

Le système réalisé, permet de construire un modèle tri-dimensionnel du type "fil de fer", en utilisant une séquence d'images fournie par une caméra fixée à la pince d'un robot manipulateur.

Sous l'hypothèse d'une scène statique, le problème peut être abordé comme l'intégration d'informations visuelles provenant de différents points de vue. Nous avons appliqué l'algorithme d'intégration sensorielle présenté dans le chapitre 5, afin de résoudre ce problème.

Dans le cadre de ces expériences, nous avons introduit et développé une technique permettant d'inférer des segments 3D, à partir d'une image bi-dimensionnelle, et de la donnée des paramètres de calibrage de la caméra.

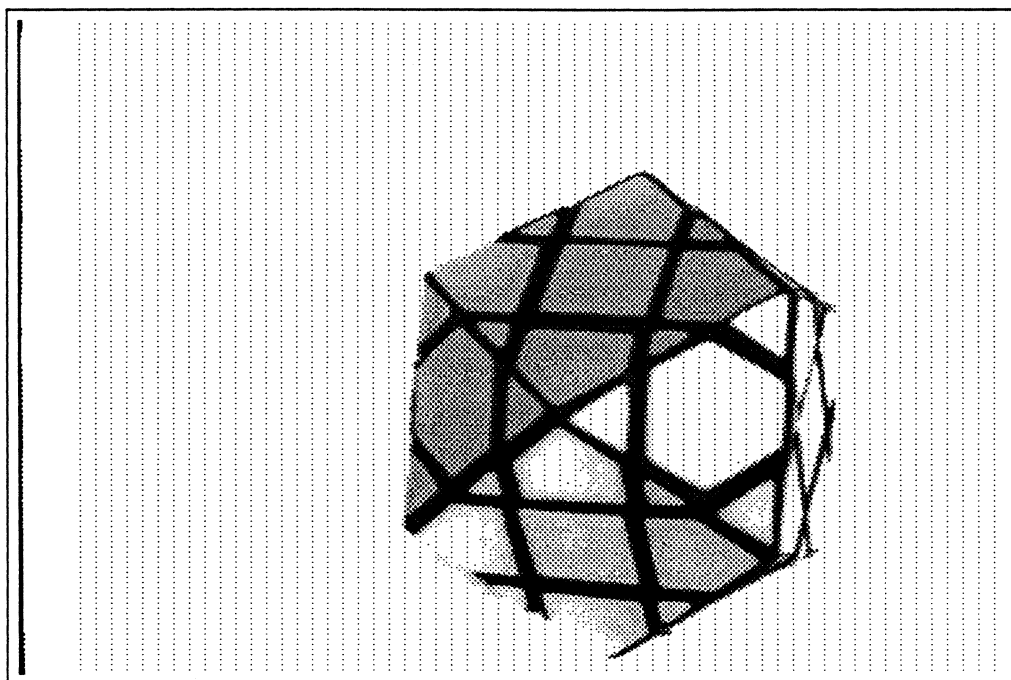


Figure 6.17: Rubik's cube

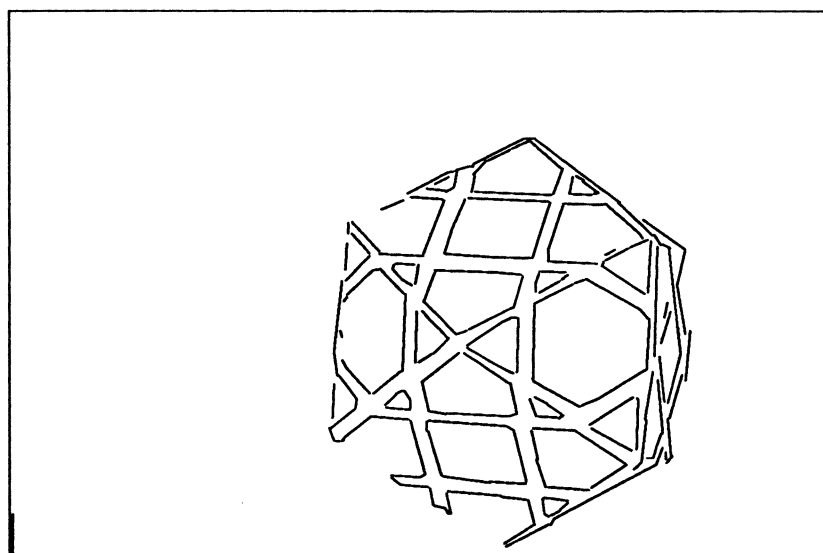


Figure 6.18: Extraction de segments 2D d'une image de rubik's cube

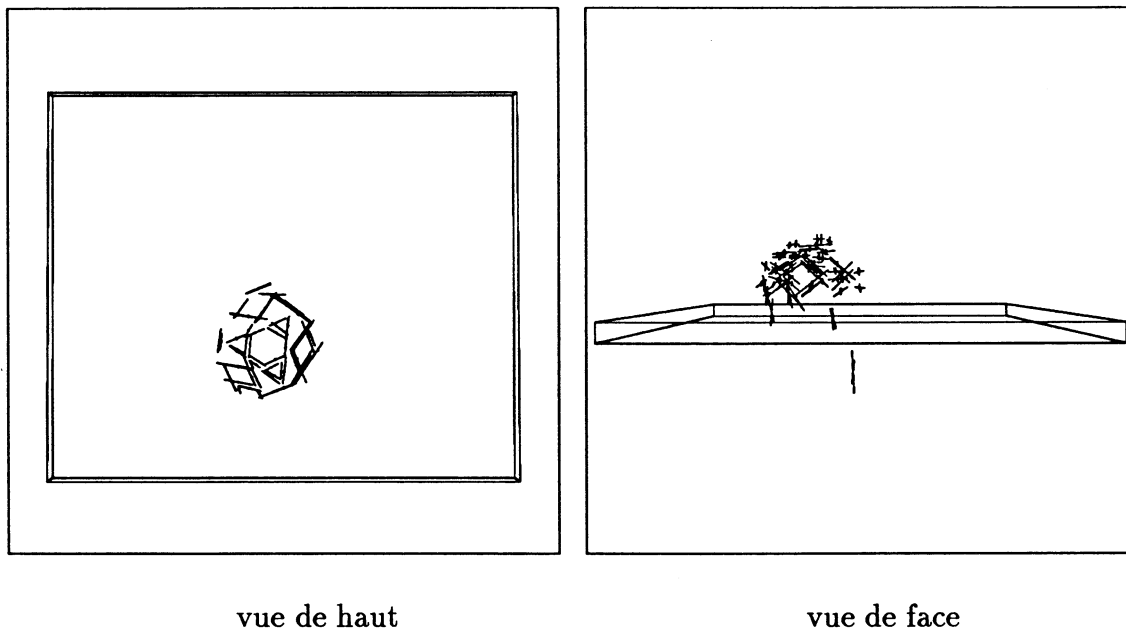


Figure 6.19: Rubiks-cube: premiers segments 3D sûrs

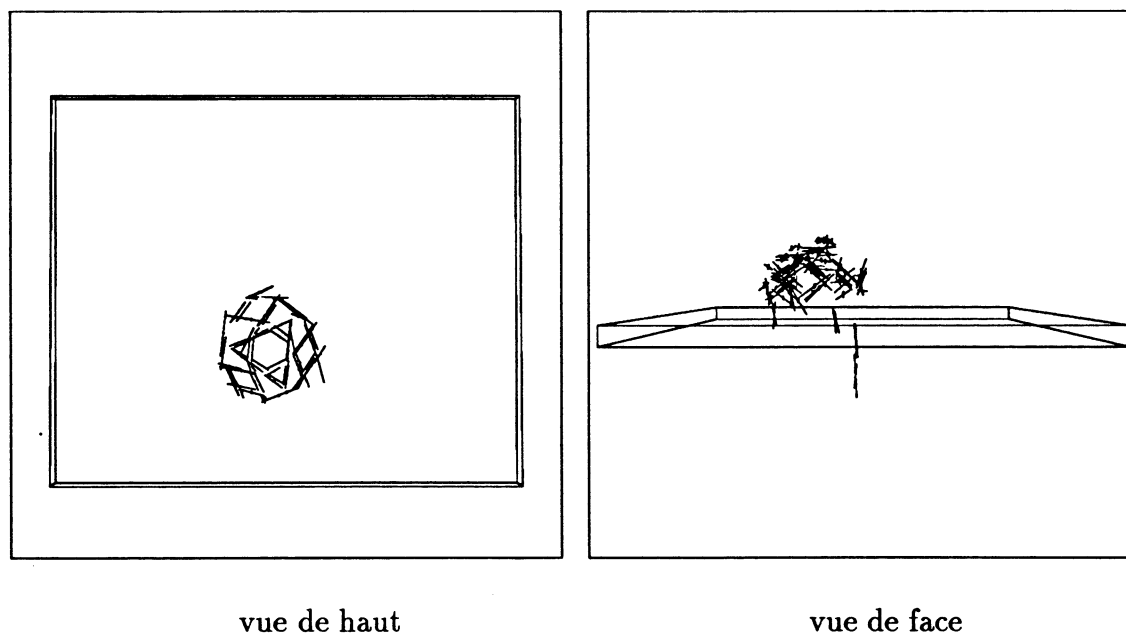


Figure 6.20: Rubiks-cube: segments 3D sûrs après huit cycles

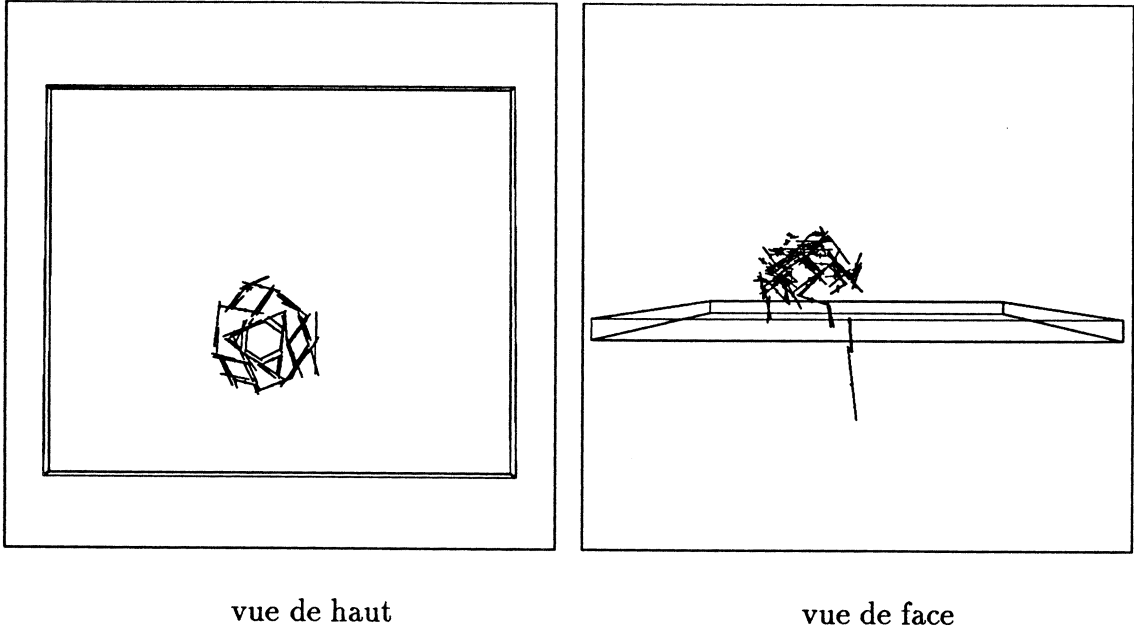


Figure 6.21: Rubiks-cube: segments 3D sûrs après dix cycles

cycle		1	2	3	4	5	6	7	8	9	10
connecteur	image	109	115	110	115	110	115	120	121	133	124
	modèle						14	17	25	25	25
rubik's cube	image	137	155	158	162	159	156	153	145	156	138
	modèle						97	111	112	115	121

Figure 6.22: Résultats des deux reconstructions



## Chapitre 7

# Application: La stéréoscopie trinoculaire

### 7.1 Principe

La stéréoscopie consiste à construire une description géométrique tri-dimensionnelle, à partir de plusieurs images simultanées de la même scène, prises en des points de vues différents et peu éloignés.

Nous avons interprété la stéréoscopie comme un processus d'intégration multisensorielle, dans lequel les caméras constituent des sources de d'informations indépendantes. Les informations géométriques extraites des trois images sont intégrées pour construire un modèle tri-dimensionnel de la scène.

Cette interprétation nous a suggéré d'utiliser notre algorithme d'intégration sensoriel pour traiter le problème de la stéréoscopie.



Figure 7.1: Triplet d'images stéréo

## 7.2 Conditions de l'expérience

Les images traitées sont présentées dans la figure 7.1. Le triplet d'images, ainsi que les paramètres intrinsèques et extrinsèques des trois caméras nous ont été fournis par l'INRIA <sup>1</sup>.

## 7.3 Construction du modèle

### 7.3.1 Traitement des images

Chacune des trois images est traitée de la façon indiquée à la section 6.3.2. Rappelons que les points de contraste sont extraits de l'image brute numérisée. Ces points sont chaînés pour former des lignes de contraste. Ces lignes sont segmentées et fournissent des segments de droite 2D (voir figure 7.2). Les informations de calibrage des caméras sont utilisées pour projeter ces segments 2D depuis l'image dans

---

<sup>1</sup>INRIA: Institut National de Recherche en Informatique et Automatique

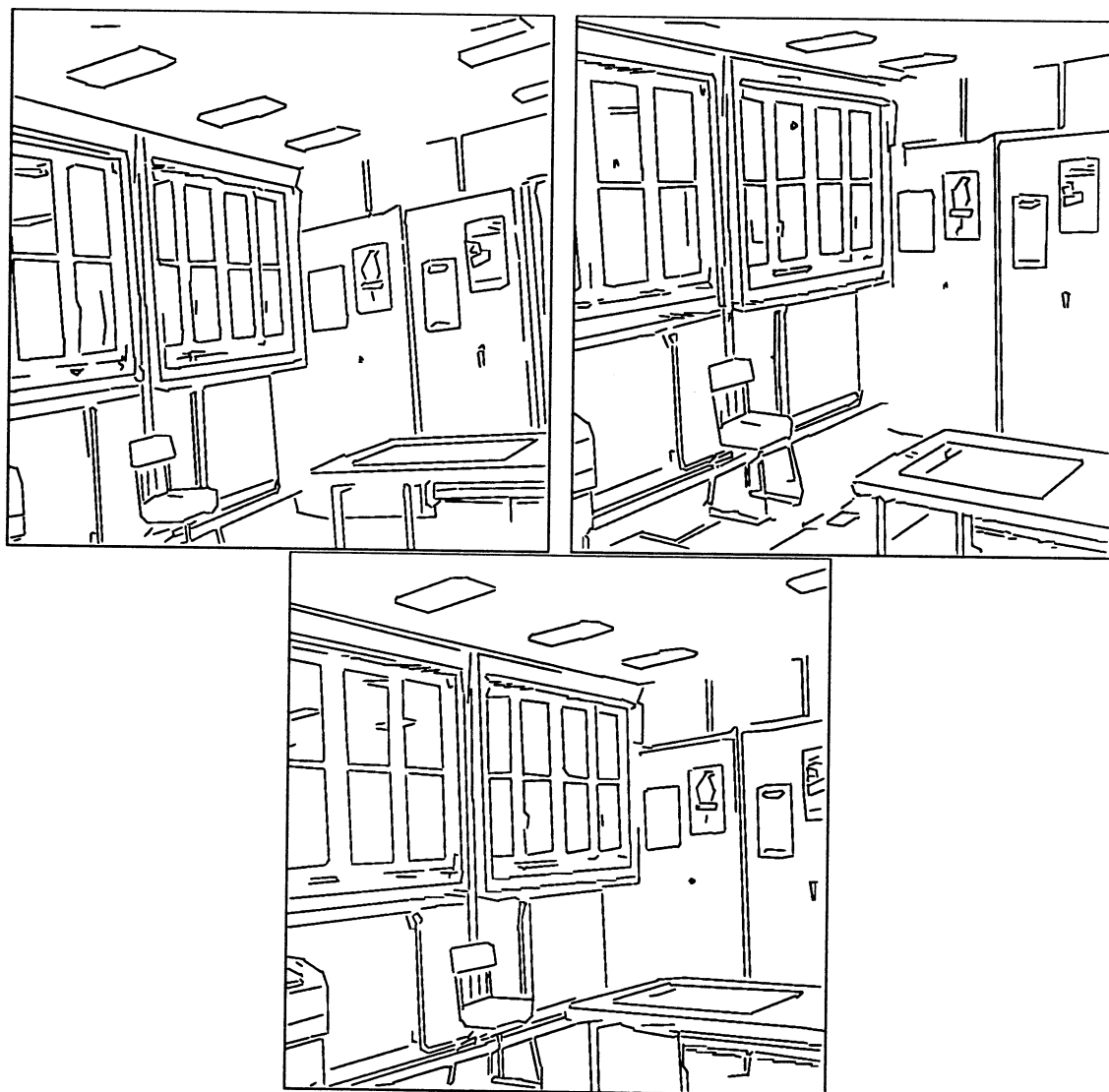


Figure 7.2: segments 2D extraits des images brutes



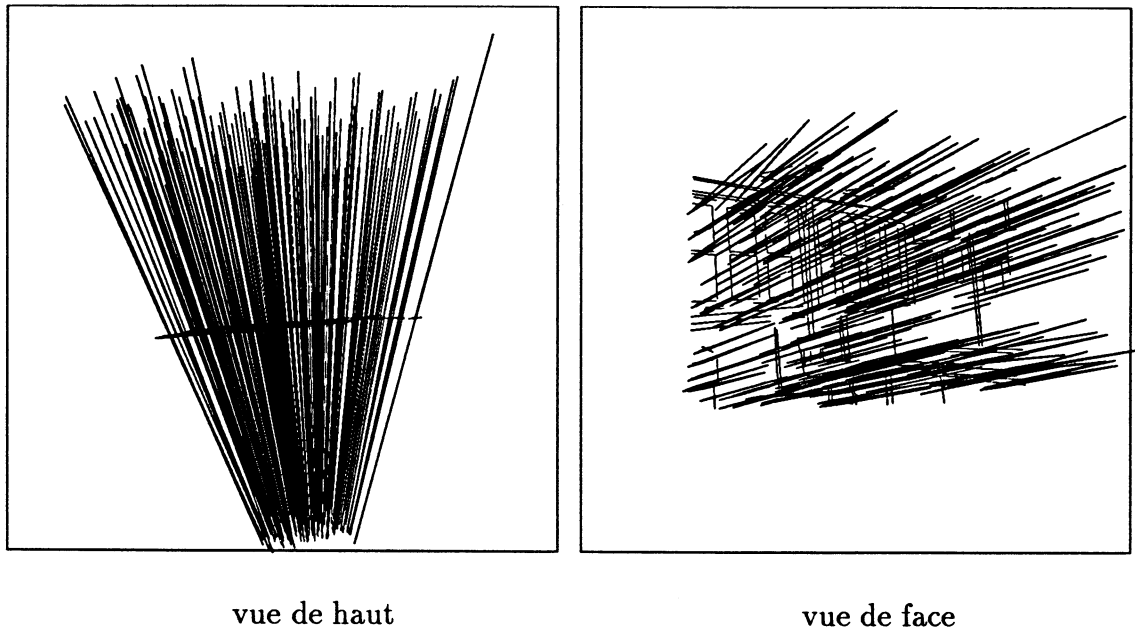


Figure 7.3: Visualisation des segments 3D provenant d'une vue, et de leur imprécision

la scène. Cette dernière opération permet donc de décrire les informations contenues dans chacune des images, en un ensemble de segments 3D affectés d'une imprécision importante dans la direction des rayons optiques (voir figure 7.3). Les segments 3D correspondants aux deux autres vues sont visualisés figure 7.4

### 7.3.2 Intégration des trois vues

Nous avons appliqué le schéma général d'intégration présenté au chapitre 5, en considérant chacune des trois vues comme une source d'information indépendante des deux autres. A chaque cycle, le processus intègre l'information provenant d'un point de vue. L'intégration totale des vues se réalise donc en trois cycles successifs. Ceci explique les valeurs que nous attribuons à  $CF_{max}$  et  $CF_{min}$ . En effet, ainsi que nous l'avons vu dans la section 5.5.2, lorsqu'un élément du modèle est confirmé par les données observées, son facteur de confiance est incrémenté d'une unité. Par conséquent, si lors de l'amorçage du processus d'intégration les hypothèses initiales sont introduites avec un facteur de confiance de 1, à l'issue des trois cycles

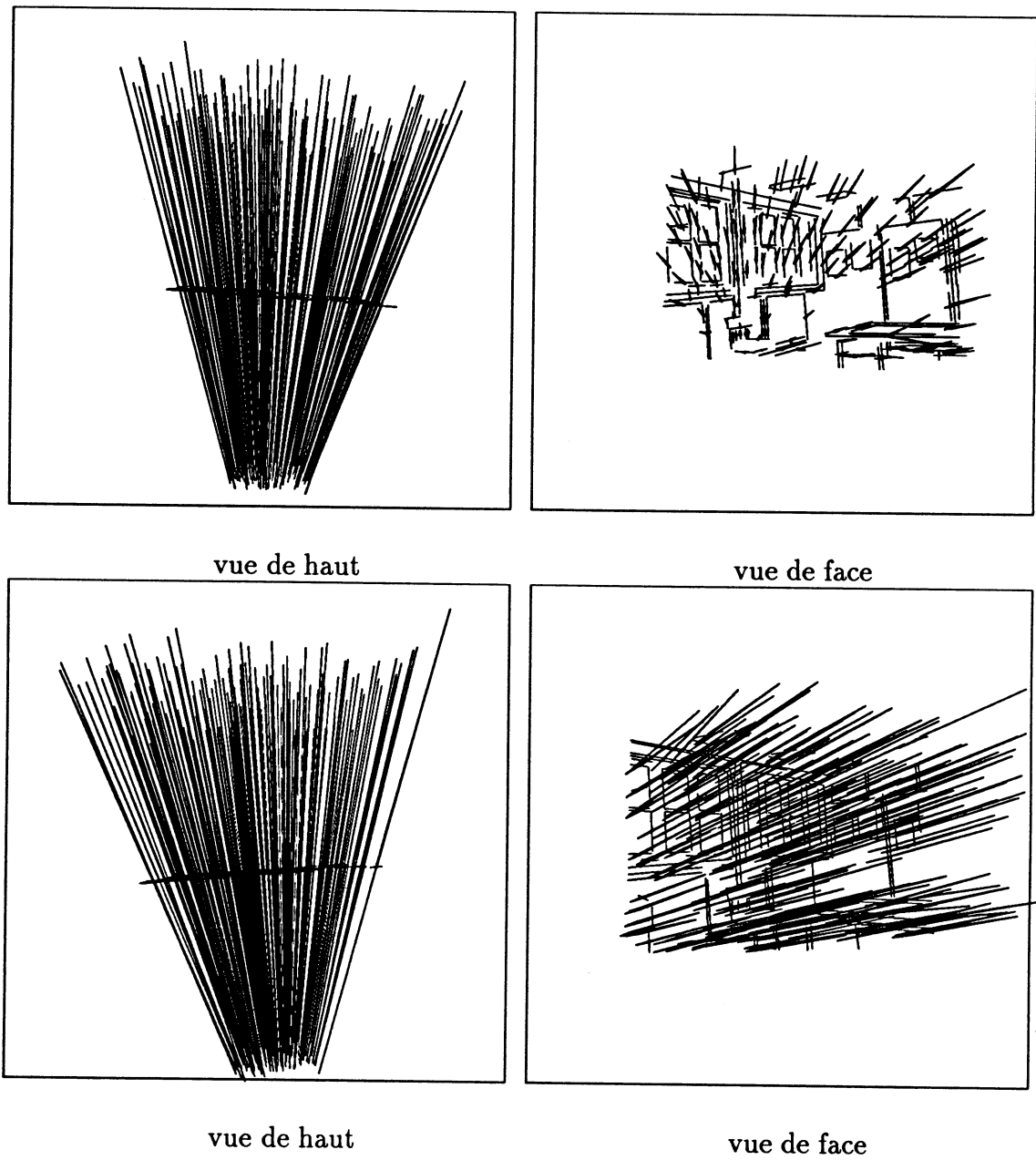


Figure 7.4: Segments 3D produits par projection inverse des deux autres vues

d'intégration, le facteur de confiance de n'importe quelle hypothèse atteindra au plus la valeur de 3. Ainsi, si l'on désire qu'à l'issue des trois cycles d'intégration le modèle puisse contenir des données sûres, il est nécessaire que  $CF_{max} = 3.0$ .

Lors de l'exécution du processus d'intégration, le système évolue de la façon suivante:

1. Lors du premier cycle, les segments 3D observés par la première caméra sont transformés en hypothèses, dont le facteur de confiance est égal à 1.
2. Lors du deuxième cycle, ces hypothèses sont confrontées aux segments 3D provenant de la deuxième caméra. Pour chaque correspondance plausible, le processus d'intégration génère un nouveau segment 3D et crée une hypothèse qui le désigne, pour indiquer que ce segment est incertain. Le processus attribue un facteur de confiance de 2 à ces nouvelles hypothèses.
3. Lors du troisième cycle, toutes les hypothèses sont à nouveau confrontées aux segments 3D provenant de la troisième caméra. Les hypothèses générées pendant ce cycle, possèdent un facteur de confiance de 3. Les hypothèses sont évaluées: si elles ne sont pas ambiguës (il n'existe pas d'autre hypothèse qui leur soit incompatible), alors elles sont validées.

Nous avons représenté figure 7.5 les segments 3D sûrs reconstruits par le processus. Le tableau 7.6 indique le nombre de segments 3D obtenus par projection inverse pour chaque image, ainsi que le nombre de segments 3D sûrs après le troisième cycle d'intégration. Ces 87 éléments sûrs sont produits en 1 heure et 2 minutes temps *cpu* sur une station SUN/260.

## 7.4 Discussion des résultats

Dans un algorithme de stéréoscopie classique, on distingue deux phases successives.

1. La mise en correspondance d'indices bi-dimensionnels entre les différentes images.
2. La triangulation, qui permet de construire des indices tri-dimensionnels à partir des indices bi-dimensionnels mis en correspondance lors de la première phase.

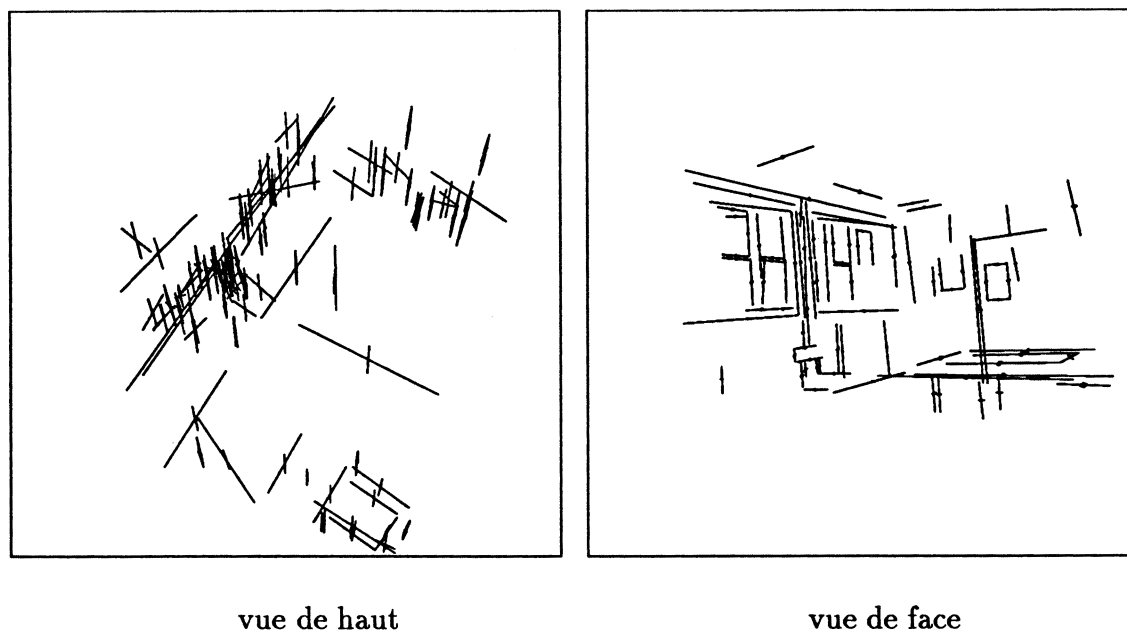


Figure 7.5: Segments 3D sûrs après trois cycles d'intégration

	vue 1	vue 2	vue 3	modèle
nombre de segments 3D	175	204	204	87

Figure 7.6: Tableau de résultats

Dans ce type de schéma, les phases de mise en correspondance et de triangulation se réalisent séquentiellement et indépendamment l'une de l'autre. Ainsi, dans un premier temps, le système cherche à mettre en correspondance les deux images ou les trois images, s'il s'agit d'un système de stéréoscopie trinoculaire. Cette mise en correspondance consiste à produire un ensemble de couples  $(i_1, i_2)$ , où  $i_1$  (resp.  $i_2$ ) est un indice bi-dimensionnel extrait de l'image 1 (resp. image 2). La difficulté de cette phase de mise en correspondance vient du nombre généralement élevé d'indices dans les images, et de l'absence d'une méthodologie permettant de trouver automatiquement parmi tous les indices de l'image 2, celui qui correspond à un indice  $i_1$  de l'image 1. En effet, autant il est facile pour un être humain d'effectuer cette mise en correspondance, autant l'élaboration d'un algorithme permettant de la réaliser de manière automatique, est une tâche ardue qui reste un sujet de recherche d'actualité.

Les différentes approches peuvent se classifier selon trois aspects:

- le choix des indices utilisés, ou plus généralement le mode de représentation des informations contenues dans une image.
- le type de contraintes utilisées pour réduire le nombre des indices de l'image 2, qui peuvent correspondre à un indice de l'image 1.
- la stratégie utilisée pour construire l'ensemble des correspondances  $(i_1, i_2)$ , à partir des correspondances potentielles autorisées par les contraintes.

Pour une description détaillée de travaux relatifs à la stéréoscopie on pourra se référer aux thèses de T. Skordas [Sko88], F. Lustman [Lus87] et N. Ayache [Aya88]..

La nécessité d'introduire les contraintes de mise en correspondance résulte de deux effets:

D'une part, la formation d'une image permet de transformer par projection, une scène tri-dimensionnelle en une description bi-dimensionnelle. Cette transformation permet de réduire l'information. Malheureusement, cette réduction est réalisée en écrasant une partie de l'information. Ainsi, deux points possédant des positions différentes dans la scène peuvent se projeter en un même point dans l'image.

D'autre part, dans un système stéréoscopique les images peuvent être prises de points de vues très différents. Par conséquent, un point donné de la scène, peut produire un point dans l'image 1, dont la position est éloignée de celle du point qu'il produit dans l'image 2.

Ces deux phénomènes contribuent à rendre le problème de mise en correspondance difficile. Sans aucune autre connaissance du système stéréoscopique, tous les indices de l'image 2 peuvent correspondre à un indice donné de l'image 1: c'est ce que l'on appelle l'ambiguïté de mise en correspondance. Afin de réduire le nombre de candidats potentiels pouvant correspondre à  $i_1$ , on est amené à utiliser des contraintes supplémentaires. Celles-ci peuvent dériver de la géométrie du système stéréoscopique (contrainte d'épipolarité), ou de connaissances sur la topologie de la scène (contrainte de continuité de la disparité ([MP77], [Gri81], [BT80], [MN85]), contrainte de gradient de disparité ([AF87b], [HAL88], [PMF85], [YP84], [BJ80]), contrainte d'unicité ([MP77]), contrainte d'ordre ([YP84]), contrainte de connectivité ([MF81], [OK85])).

L'ambiguïté de mise en correspondance trouve donc son origine dans le fait que l'information contenue dans les images est incomplète. Les contraintes permettent alors de fournir une partie de l'information manquante.

A l'opposé des systèmes classiques de stéréoscopie, notre processus de mise en correspondance s'effectue sur les indices tri-dimensionnels que constituent les segments 3D. Ainsi, l'utilisation de contraintes n'est plus nécessaire. Plus précisément, ces contraintes s'expriment implicitement dans la représentation des indices. Par exemple, la contrainte d'épipolarité entre un segment de l'image 1 et un segment de l'image 2 se traduit par un recoupement des ellipses d'incertitude des segments 3D obtenus par projection inverse.

De plus, il nous est possible d'exploiter au plus tôt l'information produite par la fusion de deux segments 3D mis en correspondance, pour contraindre d'avantage les tentatives de mise en correspondance ultérieures mettant en jeu un des deux segments 3D. Notre approche permet donc de faire coopérer les processus de mise en correspondance et de reconstruction.



## Chapitre 8

# Conclusion

### 8.1 L'évaluation de notre approche

D'un point de vue purement pratique, le système expérimental que nous avons réalisé souffre de quelques insuffisances:

- Nous n'avons pas pris en compte l'imprécision liée aux positions et déplacements de la pince. Nous avons aussi ignoré l'imprécision de calibrage de la caméra ainsi que celui du lien robot-caméra.
- Nous n'avons pas exploité les structures relationnelles définies par notre mode de représentation des informations géométriques. Ceci s'est traduit par des temps d'exécution relativement longs.

Cependant, ces insuffisances ne remettent pas en cause les thèses suivantes que nous voulions défendre:

Ces thèses sont les suivantes:



1. Pour envisager un système d'intégration multisensorielle, il est nécessaire de représenter explicitement la précision des informations géométriques.
2. Pour s'affranchir des particularités propres aux capteurs, ou aux techniques d'inférence de forme, il est pratique d'utiliser une représentation uniforme des données.
3. Pour prendre en compte le manque de fiabilité de certains capteurs, il est nécessaire de représenter explicitement l'incertitude des informations.

Nous avons proposé des solutions à chacun de ces points. Celles-ci présentent les caractéristiques suivantes:

Notre représentation géométrique procure deux avantages: D'une part, elle contient une source importante d'information notamment dans des tâches descendantes, ou dirigées par les buts, comme la reconnaissance d'objets.

D'autre part, nous avons montré que cette structure relationnelle pouvait faciliter le contrôle de la mise en correspondance, se guidant localement d'une façon hiérarchique. Cela permet de diminuer considérablement la complexité des phases de mise en correspondance des données que comprend le processus d'intégration.

Notre représentation de la précision à l'aide des variables aléatoires multi-dimensionnelles est concise. Elle permet de présenter de façon cohérente et sous un même formalisme, différents outils de manipulation de la précision.

Nos résultats expérimentaux ont permis de valider notre approche.

## 8.2 Perspectives futures

### 8.2.1 Les autres capteurs?

Nous avons développé nos outils et notre méthodologie, avec un constant souci de généralité. Nous les avons appliqués à un type particulier de capteur (la vision monoculaire), mais ceux-ci peuvent être utilisés directement et sans trop d'effort, pour combiner les informations fournies par différents capteurs tels que:

- les capteurs à ultra-sons.

- les nappes lasers.
- les capteurs tactiles.

Notre expérience montre que le travail le plus important et le plus difficile consiste à calibrer les capteurs de façon correcte.

### 8.2.2 Le modèle composite: pour quoi faire?

Une modélisation géométrique n'est utile, que si l'on s'en sert. Pour compléter la validation de notre approche, il est nécessaire de tester l'efficacité et l'adéquation du modèle composite à informer les modules en amont de la perception, tels que la reconnaissance, la navigation, la manipulation d'objets ou l'interprétation de scène.

### 8.2.3 Vers une description plus riche

Les primitives de la modélisation géométrique que nous avons définies, conviennent parfaitement aux objets et surfaces polyédriques. Pour adapter le modèle composite à des environnements réels variés, il nous semble important d'enrichir les classes de primitives géométriques afin d'étendre notre vocabulaire de description. Pour cela, il convient d'introduire des primitives surfaciques plus évoluées telles que les sphères, les ellipsoïdes, les cônes. Le choix des primitives géométriques devrait essentiellement dépendre de notre capacité à manipuler efficacement ces primitives.

### 8.2.4 Un univers dynamique

Dans notre étude, nous nous sommes restreints à deux situations qui définissent la même problématique:

- des acquisitions simultanées d'informations par plusieurs capteurs.
- des acquisitions successives, dans le contexte d'une scène statique.

Pour traiter le cas de scène dynamique, il est nécessaire de faire intervenir le temps, le déplacement, ainsi que la vitesse des objets. La représentation de ces notions et leur interaction avec la modélisation géométrique, nécessite un effort de recherche important, mais indispensable.



# Liste des Figures

1.1	Le rôle de la perception: construire le modèle interne de l'environnement	7
1.2	Types de données manipulées par 3D-FORM [E.L.Walker & al. A framework for representing and reasoning about three-dimensional objects for vision. AI Magazine n.9, 1988]	11
1.3	Grille de certitude [H.P.Moravec, Sensor fusion in certainty grids for mobile robots. AI Magazine n.9, 1988]	12
1.4	Types de données manipulées par le système de l'INRIA [N.Ayache & al. Maintaining representations of the environment of a mobile robot. Technical Report 789, INRIA Rocquencourt, 1988]	15
1.5	schéma fonctionnel du processus d'intégration	23
2.1	Calcul de $P([a, b])$ à partir de la courbe de densité de probabilité	28
2.2	Densité gaussienne centrée réduite.	28
2.3	surface de densité gaussienne [J.L. Crowley, A representation for visual information, PhD thesis CMU, nov.1981]	31
2.4	ellipse d'incertitude d'une v.a.m. bi-dimensionnelle	32
2.5	Comparaison de deux v.a.m. de dimension 2	36
2.6	Combinaison de deux v.a.m. de dimension 2	39
2.7	Réestimation de deux v.a.m. de natures différentes	40

3.1	Représentation par échantillonnage de l'espace [D.H.Ballard and C.M.Brown, Computer vision, Prentice Hall 1982] . . . . .	47
3.2	Représentation par volumes [A.G. Requicha and H.B. Voelckert, Solid Modelling: a historical summary and contemporary assessment, IEEE Computer Graphics and its applications, 1982] . . . . .	48
3.3	Représentation par surfaces [A.G. Requicha and H.B. Voelckert, Solid Modelling: a historical summary and contemporary assessment, IEEE Computer Graphics and its applications, 1982] . . . . .	48
3.4	Sommet 3D . . . . .	51
3.5	Segment 3D . . . . .	52
3.6	Élément de surface . . . . .	53
3.7	exemple de graphe de relations géométriques . . . . .	56
4.1	Structure d'une hypothèse . . . . .	64
4.2	Réseau de dépendance des hypothèses . . . . .	65
4.3	Algorithme de réévaluation d'une hypothèse . . . . .	66
4.4	réévaluation d'une hypothèse . . . . .	67
4.5	Situation après réévaluation de $a$ . . . . .	68
4.6	Algorithme de destruction d'une hypothèse . . . . .	69
4.7	Situation après destruction de $a'$ . . . . .	70
5.1	segments 3D se correspondant . . . . .	79
5.2	Fusion de deux segments 3D: construction des extrémités . . . . .	81
5.3	Algorithme de fusion de deux segments 3D . . . . .	82
5.4	primitives 3D du modèle . . . . .	85
5.5	primitives 3D observées . . . . .	86
5.6	Algorithme de mise en correspondance globale . . . . .	87
5.7	Algorithme de mise en correspondance de primitives . . . . .	88
5.8	Processus cyclique d'intégration . . . . .	90
6.1	Montage caméra-robot manipulateur . . . . .	94
6.2	Architecture matérielle . . . . .	95
6.3	Calcul du lien caméra-robot . . . . .	98
6.4	Configuration de l'expérience . . . . .	102

6.5	Image en niveau de gris . . . . .	102
6.6	Extraction des contours . . . . .	103
6.7	Extraction de segments 2D . . . . .	104
6.8	Projection inverse des segments images. Vue de dessus . . . . .	104
6.9	Vue d'un segment 3D et de l'imprécision du point milieu . . . . .	105
6.10	Projection inverse des segments images, vue de côté . . . . .	106
6.11	Etat initial du système . . . . .	107
6.12	segments 3D générés lors du second cycle . . . . .	110
6.13	Correspondances incompatibles entre segments 3D . . . . .	110
6.14	Premiers segments 3D sûrs . . . . .	111
6.15	Eléments sûrs après 8 cycles d'intégration . . . . .	111
6.16	Eléments sûrs après 10 cycles d'intégration . . . . .	112
6.17	Rubik's cube . . . . .	115
6.18	Extraction de segments 2D d'une image de rubik's cube . . . . .	115
6.19	Rubiks-cube: premiers segments 3D sûrs . . . . .	116
6.20	Rubiks-cube: segments 3D sûrs après huit cycles . . . . .	116
6.21	Rubiks-cube: segments 3D sûrs après dix cycles . . . . .	117
6.22	Résultats des deux reconstructions . . . . .	117
7.1	Triplet d'images stéréo . . . . .	120
7.2	segments 2D extraits des images brutes . . . . .	121
7.3	Visualisation des segments 3D provenant d'une vue, et de leur imprécision	122
7.4	Segments 3D produits par projection inverse des deux autres vues .	123
7.5	Segments 3D sûrs après trois cycles d'intégration . . . . .	125
7.6	Tableau de résultats . . . . .	125
B.1	Calcul du lien caméra-robot . . . . .	148
B.2	Recherche de la position de montage d'une caméra, par résolution d'une équation de la forme: $AX = XB$ , où A est le déplacement du robot, B est le déplacement de la caméra, et X est la position de montage de la caméra. . . . .	149



# Bibliographie

- [AB85] G.J. Agin and T.O. Binford. Computer description of curved objects. *IEEE Transaction on Computer*, 1985.
- [AF87a] N. Ayache and O. D. Faugeras. Building, registering, and fusing noisy visual maps. In *Proc. First International Conference on Computer Vision*, pages 73–82, London, England, June 1987.
- [AF87b] N. Ayache and B. Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1(2):107–131, 1987.
- [AF88] N. Ayache and O. D. Faugeras. *Maintaining Representations of the Environment of a Mobile Robot*. Technical Report 789, INRIA Rocquencourt, France, February 1988.
- [Als83] D. L. Alspach. An approach to the multi-sensor integration problem. In *EASCON Proc. of the 16th. annual IEEE conference on electronics and aerospace systems*, pages 149–152, Washington, DC, 1983.
- [Aya88] N. Ayache. *Construction et Fusion de Représentations Visuelles 3D, Applications à la Robotique Mobile*. Thèse d'état, Université de Paris-Sud, Centre d'Orsay, 1988.



- [BB82] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982.
- [BGB79] R. A. Brooks, R. Greiner, and T. O. Binford. The acronym model-based vision system. In *Proc. of International Conference of Artificial Intelligence*, Tokyo, august 1979.
- [BJ80] P. Burt and B. Julesz. A disparity gradient limit for binocular fusion. *Science*, 208:615–617, May 1980.
- [Bro82] R. A. Brooks. Symbolic error analysis and robot planning. *Int. Jour. of Robotics Research*, 1(4):29–68, winter 1982.
- [Bro85] R. A. Brooks. Aspects of mobile robot visual map making. In *IEEE Int. Conf. on Robotics and Automation*, St Louis, USA, March 1985.
- [BT80] S. T. Barnard and W. B. Thompson. Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(4):333–340, July 1980.
- [Can83] J. F. Canny. *Finding edges and lines in images*. Technical Report TR-720, M.I.T. Cambridge, 1983.
- [Cro85] J.L. Crowley. The construction and maintenance of a composite surface model. In *SPIE Proc.*, 1985.
- [Cro86] J.L. Crowley. Navigation and world modeling for a mobile robot: a progress report. In *Proc. of the 3rd International Joint Conference on CAD and Robotics in Construction*, 1986.
- [Dem86] Y. Demazeau. *Niveaux de représentation en vision par ordinateur : indices d'images et indices de scène*. PhD thesis, Institut National Polytechnique de Grenoble, France, December 1986.
- [Der87] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.
- [dK86] J. de Kleer. An assumption-based tms. *Artificial Intelligence*, 28(2):127–162, 1986.

- [Doy77] J. Doyle. *Truth Maintenance systems for problem solving*. PhD thesis, M.I.T. Cambridge, Cambridge, Massachusetts, 1977.
- [Dur86] H. F. Durrant-Whyte. *Integration, Coordination and Control of Multi-Sensor Robot Systems*. PhD thesis, University of Pennsylvania, GraspLab, 1986.
- [Dur87] H. F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *International Journal of Robotics Research*, 6(3):3–24, Fall 1987.
- [Fau82] O.D. Faugeras. 3-d shape representation. In O.D. Faugeras, editor, *Fundamentals in Computer Vision*, pages 293–303, Cambridge University Press, 1982.
- [FP85] H. Farreny and H Prade. Tackling uncertainty and imprecision in robotics. In *3rd Int. Symp. in Robotics Research*, Gouvieux, France, 1985.
- [FT86] O. D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proc. Computer Vision and Pattern Recognition*, pages 15–20, Miami Beach, Florida, USA, June 1986.
- [Gri81] W. E. L. Grimson. *From Images to Surfaces*. MIT Press, Cambridge, Massachusetts, 1981.
- [HAL88] C. Hansen, N. Ayache, and F. Lustman. Towards real-time trinocular stereo. In *2th. ICCV Proc.*, pages 129–133, Tampa, Florida, December 1988.
- [HK86] M. Herman and T. Kanade. Incremental reconstruction of 3D scenes from multiple, compleximages. *Artificial Intelligence*, 30:289–341, 1986.
- [HW77] D. Hubel and T. Wiesel. Functional architecture of macaque monkey visual cortex. In *Proceedings of the Royal Society*, pages 1–59, London, 1977.
- [Kam85] S. J. Kamat. Value function structure for multiple sensor integration. In *Proc. SPIE, Intelligent robots and computer vision*, pages 432–435, Cambridge, Ma., September 1985.

- [LKG78] M. Labarrere, J. P. Krief, and B. Gimonet. *Le filtrage et ses applications*. Cepadus edition, Sup'aero, 1978.
- [Lus87] F. Lustman. *Vision Stéréoscopique et Perception du Mouvement en Vision Artificielle*. PhD thesis, Université de Paris-Sud Centre d'Orsay, France, December 1987.
- [MA86] A. Mitiche and J.K. Aggarwal. Multiple sensor integration/fusion through image processing: a review. *Optical Engineering*, 1(3):380–386, March 1986.
- [Mar82] D. Marr. *Vision*. W. H. Freeman, San Francisco, 1982.
- [Met72] M. Metivier. *Notions Fondamentales de la Theorie des Probabilites*. Dunod, 1972.
- [MF81] J. E. W. Mayhew and J. P. Frisby. Psychophysical and computational studies towards a theory of human stereopsis. *Artificial Intelligence*, 17:349–385, 1981.
- [MN85] G. Medioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics, and Image Processing*, 31:2–18, 1985.
- [Mor88] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, summer 1988.
- [MP77] D. Marr and T. Poggio. *A Theory of Human Stereo Vision*. Technical Report AI Memo 451, MIT Artificial Intelligence Laboratory, November 1977.
- [Nil65] N. Nilsson. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*. McGraw-Hill, 1965.
- [OK85] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154, March 1985.
- [Per86] J. Pertin. *Modélisation du raisonnement géométrique pour la programmation des robots*. PhD thesis, INPG, Grenoble, France, 1986.

- [PMF85] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.
- [Por87] J. Porrill. Optimal combination and constraints for geometrical sensor data. rapport interne, March 1987.
- [Pug88] P. Puget. La vérification/correction de programme au sein d'un système de programmation automatique de robots. In *Journées Géométrie et Robotique*, CNRS-INRIA, LAAS, Toulouse, mai 1988.
- [RC87] F. Ramparany and J.L. Crowley. Representation et manipulation de l'incertitude geometrique pour la perception. In *Actes du congrès RFIA*, 1987.
- [RdSV86] A. Robert de Saint Vincent. *Perception et Modélisation de l'Environnement d'un Robot Mobile : Une Approche par Stéréovision*. PhD thesis, Université Paul Sabatier de Toulouse, November 1986.
- [RV79] A. G. Requicha and H. B. Voelker. Geometric modelling of mechanical parts and machining processes. *COMPCONTROL*, November 1979.
- [RV82] A. G. Requicha and H. B. Voelckert. Solid modelling: a historical summary and contemporary assesement. *IEEE, Computer Graphics and its applications*, march 1982.
- [SA86] Y.C. Shiu and S. Ahmad. Finding the mounting position of a sensor by solving a homogeneous transform equation of the form  $ax = xb$ . In *Proc. IEEE International Conference of Robotics and Automation*, pages 1666–1671, San Francisco, California, USA, June 1986.
- [SC86] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, Winter 1986.
- [Sko88] Th. Skordas. *Mise en correspondance et reconstruction stéréo utilisant une description structurelle des images*. PhD thesis, Institut National Polytechnique de Grenoble, September 1988.

- [SKS86] S. Shekhar, O. Khatib, and M. Shimojo. Sensor fusion and object localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 1623–1628, San Francisco, Ca., april 1986.
- [SST86] S. A. Shafer, A. Stentz, and C. E. Thorpe. An architecture for sensor fusion in a mobile robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 2002–2011, San Francisco, Ca., april 1986.
- [WHK88] E. L. Walker, M. Herman, and T. Kanade. A framework for representing and reasoning about three-dimensional objects for vision. *AI Magazine*, 9(2):47–58, summer 1988.
- [YP84] A. L. Yuille and T. Poggio. *A Generalized Ordering Constraint for Stereo Correspondence*. Technical Report AI Memo 777, MIT Artificial Intelligence Laboratory, May 1984.
- [Zuc85] S. W. Zucker. Early processes for orientation selection and grouping. In A. P. Pentland, editor, *From Pixels to Predicates*, pages 170–200, Ablex Publishing Corporation, 1985.

## Annexe A

# Calibrage de la caméra

La technique de calibrage présentée est celle développée dans l'article [FT86].

Les auteurs modélisent la transformation opérée par la caméra, par une projection perspective. Dans le repère de la caméra <sup>1</sup>, les coordonnées  $(X_1, Y_1, Z_1)$  d'un point de la scène, sont reliées aux coordonnées  $(u, v)$  de son point image, par la relation:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{bmatrix} k_u f & 0 & u_0 & 0 \\ 0 & k_v f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{pmatrix} \quad (\text{A.59})$$

où:

$(u_0, v_0)$  sont les coordonnées pixels du centre de l'image.

$f$  est la distance focale.

---

<sup>1</sup>Le repère de la caméra est défini par  $(O_1, X_1, Y_1, Z_1)$ , où  $O_1$  est le centre focal de la caméra,  $Z_1$  est la direction focale et  $X_1$  et  $Y_1$ , sont les directions de la rétine.

$k_u, k_v$  sont les facteurs d'échelle, entre les unités de mesure sur l'image et les unités de mesure dans la scène:  $\Delta X = k_u \Delta u$ . ( $\Delta X$  est exprimée en *mm*,  $\Delta u$  est exprimée en pixels.)

$s$  est la coordonnée homogène.

La relation A.59 est de la forme:

$$U = MX \quad (\text{A.60})$$

où  $M$  est appelée matrice de projection perspective.

Pour exprimer cette matrice  $M$  dans le repère de la scène, nous devons effectuer un changement de coordonnées, par transformation rigide.

Dans le repère de la scène, la matrice  $M$  s'écrit alors:

$$M' = \begin{bmatrix} k_u f r_1 + u_0 r_3 & k_u f t_x + u_0 t_z \\ k_v f r_2 + v_0 r_3 & k_v f t_y + v_0 t_z \\ r_3 + t_z \end{bmatrix} \quad (\text{A.61})$$

où:  $y$

$r_i$  est le  $i^{\text{ième}}$  vecteur ligne de la matrice de rotation de la transformation rigide.

$(t_x, t_y, t_z)$  sont les coordonnées de la translation de la transformation rigide.

Par la suite, on utilise la notation:

$$M' = \begin{bmatrix} I_1 & I_{14} \\ I_2 & I_{24} \\ I_3 & I_{34} \end{bmatrix}$$

La matrice  $M'$  relie les coordonnées  $X_w$  points de la scène (dans le repère de la scène), à leurs coordonnées image dans le plan rétinien par la relation:

$$U = M' X_w \quad (\text{A.62})$$

En utilisant les propriétés des matrices de rotation (matrices orthonormées), et la forme particulière de la matrice de projection perspective, les auteurs dégagent

un ensemble de relations qui permettront par la suite, de calculer les paramètres intrinsèques et extrinsèques à partir des coefficients de la matrice  $M'$ .

Ces relations s'écrivent:

$$\begin{aligned}
 I_3 I_3^t r &= k^2 & k^2 u_0 v_0 &= I_1 I_2^t r \\
 k^2 u_0 &= I_1 I_3^t r & k^2 (\alpha_v^2 + v_0^2) &= I_2 I_2^t r \\
 k \alpha_u r_1 &= I_1 - u_0 I_3 & k \alpha_v r_2 &= I_2 - v_0 I_3 \\
 k r_3 &= I_3 & & \\
 k^2 (\alpha_u \alpha_v r_3 - \alpha_u v_0 r_2 - \alpha_v u_0 r_1) &= I_1 \times I_3 & & \\
 k^2 \alpha_u r_2 &= I_3 \times I_1 & k^2 \alpha_v r_1 &= I_2 \times I_3 \\
 k \alpha_u t_x &= I_{14} - u_0 I_{34} & k \alpha_v t_y &= I_{24} - v_0 I_{34} \\
 k t_z &= I_{34} & & 
 \end{aligned} \tag{A.63}$$

Pour calculer les coefficients de la matrice  $M'$ , les auteurs proposent d'utiliser un ensemble de points particuliers de la scène, dont on connaît les coordonnées scène  $X_w$  et les coordonnées dans l'image  $U$ . En écrivant la relation A.62 pour ces  $n$  points particuliers, on obtient le système d'équations:

$$M' X_{wi} = U_i, i \in \{1, \dots, n\} \tag{A.64}$$

Dès lors que  $n > 12$ , le système A.64 peut être considéré comme un système d'équations linéaires surdéterminé, dont les inconnues seraient les coefficients de la matrice  $M'$ .

Ces coefficients sont calculés au moyen de la technique des moindres carrés.

Une fois  $M'$  calculée, on peut utiliser les relations A.63 pour retrouver la valeur des paramètres intrinsèques et extrinsèques.





## Annexe B

# Calibrage du lien robot-caméra

Ce calibrage permet de calculer la matrice de passage, permettant de passer du repère de la pince du robot au repère de la caméra (figure B.1). La technique que nous utilisons s'inspire des travaux présentés dans l'article [SA86].

Lorsque la pince se déplace d'une position 1 à une position 2, elle déplace également la caméra. Ces deux déplacements peuvent évaluer indépendamment l'un de l'autre:

Le déplacement de la pince peut s'évaluer ainsi: le contrôleur du bras fournit la matrice de passage  $T_6$ , permettant de passer du repère station du robot au repère de la pince. Ainsi, le contrôleur fournit deux matrices de passage  $T_{6_1}$  et  $T_{6_2}$  correspondant aux deux positions. La matrice  $A$  caractérisant le déplacement de la pince de la position 1 à la position 2, s'écrit alors:

$$A = T_{6_1}^{-1}T_{6_2} \tag{B.65}$$

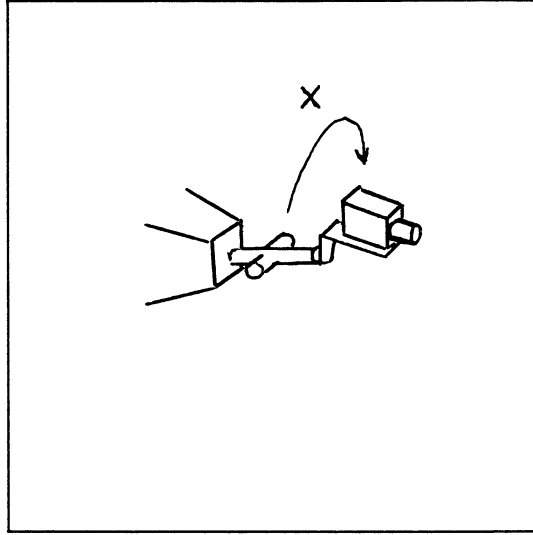


Figure B.1: Calcul du lien caméra-robot

Le déplacement de la caméra peut s'évaluer, si dans chacune des deux positions 1 et 2, on effectue un calibrage de la caméra. Celui-ci est réalisée à l'aide de la technique présentée dans le paragraphe 6.3.1.

Dans la position 1, un calibrage de la caméra fournit la matrice de passage  $C_1$ , qui permet de passer du repère de la caméra en position 1, au repère de la scène. De même, un calibrage de la caméra dans la position 2, fournit la matrice de passage  $C_2$ , qui permet de passer du repère de la caméra en position 2, au repère de la scène. La matrice  $B$  caractérisant le déplacement de la caméra de la position 1 à la position 2 s'écrit alors:

$$B = C_1 C_2^{-1} \quad (\text{B.66})$$

Les deux déplacements sont contraints, par le fait que la caméra est fixée à la pince à l'aide d'une liaison rigide. La matrice  $X$  que nous cherchons, caractérise cette liaison. La contrainte entre les deux déplacements se traduit par l'équation:

$$AX = XB \quad (\text{B.67})$$

Celle-ci est bien illustrée par la figure B.2. Les auteurs de l'article démontrent les trois résultats suivants:

1. L'équation B.67 admet une infinité de solutions, dans l'espace des transformations homogènes.

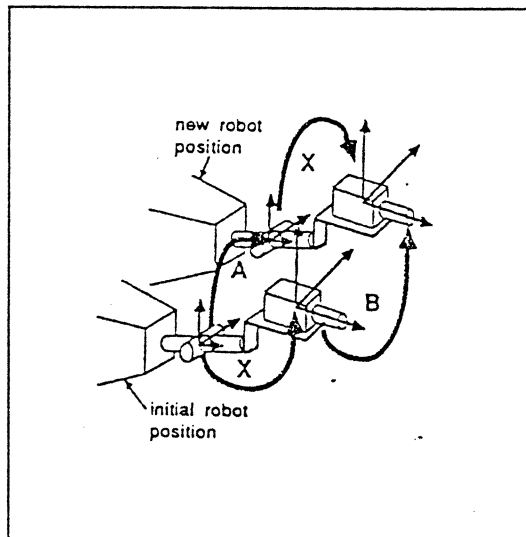


Figure B.2: Recherche de la position de montage d'une caméra, par résolution d'une équation de la forme:  $AX = XB$ , où  $A$  est le déplacement du robot,  $B$  est le déplacement de la caméra, et  $X$  est la position de montage de la caméra.

2. Pour trouver la solution  $X$ , il est nécessaire de réaliser deux déplacements de la pince, tels que les axes des deux rotations ne soient pas parallèles. La matrice  $X$  est alors solution du système d'équations:

$$\begin{cases} A_1X = XB_1 \\ A_2X = XB_2 \end{cases} \quad (\text{B.68})$$

où  $A_1, A_2$  (resp.  $B_1, B_2$ ) sont les matrices homogènes des deux déplacements de la pince (resp. caméra)

3. La solution  $X$  de ce système d'équations existe, et est unique. En outre, l'article propose une méthode permettant de calculer la solution, en utilisant explicitement le fait que  $X$  est une transformation homogène.

Nous avons défini une procédure de calibrage du lien robot-caméra, mettant en œuvre les principes développés dans l'article [SA86]. Cette procédure commande au robot de placer la pince dans trois positions prédéfinies. A chacune des positions, on effectue un calibrage de la caméra. Les résultats des calibrages, ainsi que les positions de la pince, sont traités pour calculer la matrice  $X$  cherchée.



## Annexe C

# Technique de la projection inverse

Nous détaillons les différentes étapes de la projection inverse:

Soient  $Z_{min}$  et  $Z_{max}$  les profondeurs minimales et maximales des points, vus par la caméra. La variable aléatoire de moyenne  $Z_0 = \frac{Z_{min}+Z_{max}}{2}$ , et de variance  $\sigma_{Z0} = \frac{(Z_{min}-Z_{max})^2}{4}$ , représente la distribution des profondeurs possibles des points visibles.

Soient  $(x_i, y_i)$  les coordonnées du point image. L'échantillonnage des images, induit une incertitude sur ces coordonnées, que l'on peut modéliser par une variable aléatoire bi-dimensionnelle de moyenne  $(x_i, y_i)$  et de matrice de covariance:

$$\begin{bmatrix} \sigma_{x_i}^2 & 0 \\ 0 & \sigma_{y_i}^2 \end{bmatrix}$$

(où  $\sigma_{x_i}$  et  $\sigma_{y_i}$  sont les dimensions du pixel.)

Les équations de la projection perspective, permettent de relier la profondeur aléatoire

et les coordonnées aléatoires du point image, aux coordonnées aléatoires du point 3D. Ces équations s'écrivent:

$$\begin{cases} X = \frac{xZ_0}{F} \\ Y = \frac{yZ_0}{F} \\ Z = Z_0 \end{cases} \quad (C.69)$$

elles fournissent directement la moyenne  $(X, Y, Z)$ , du point aléatoire 3D. Pour obtenir sa matrice de covariance  $\Lambda_P$ , il faut dans un premier temps calculer la matrice jacobienne  $J$  des relations C.69, dans lesquelles les coordonnées  $(X, Y, Z)$  sont considérées comme des fonctions des variables aléatoires  $Z_0, x$  et  $y$ . La matrice jacobienne  $J$  s'écrit:

$$M = \begin{bmatrix} \frac{Z_0}{F} & 0 & \frac{x}{F} \\ 0 & \frac{Z_0}{F} & \frac{y}{F} \\ 0 & 0 & 1 \end{bmatrix} \quad (C.70)$$

Les coordonnées des points et des vecteurs, ainsi que les matrices de covariance associées, sont rapportées au repère de la caméra.

Pour calculer les coordonnées de ces mêmes points et vecteurs dans le repère absolu, nous devons appliquer un changement de repère.

Soient  $(X', Y', Z')$ , les coordonnées dans le repère absolu. Celles-ci sont liées aux coordonnées  $(X, Y, Z)$  par les formules:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R_6 R_X \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + R_6 t_X + t_6 \quad (C.71)$$

où:

$R_6$  et  $t_6$  sont la rotation et la translation qui définissent la transformation entre le repère station, et le repère de la pince.  $R_X$  et  $t_X$  sont la rotation et la translation qui définissent la position du repère de la caméra, par rapport au repère de la pince.

La matrice de covariance  $\Lambda'$  des coordonnées  $(X', Y', Z')$  peut être obtenue en appliquant la technique présentée au paragraphe 2.4.3. Soit:

$$\Lambda' = R_6 R_X \Lambda R_X^T R_6^T \quad (C.72)$$

Pour appliquer l'opération de projection inverse à des segments 2D, nous l'appliquons successivement aux deux points qui forment les extrémités du segment 2D. Nous formons un segment 3D à partir des deux points 3D obtenus.

Les paramètres définis au chapitre 3 sont calculés ainsi:

$$P = \frac{P_1 + P_2}{2}$$

$$L = \|\overrightarrow{P_1 P_2}\|$$

$$N = \frac{\overrightarrow{P_1 P_2}}{\|\overrightarrow{P_1 P_2}\|}$$

$$\Lambda_P = \frac{1}{4}(\Lambda_{P_1} + \Lambda_{P_2})$$

$$\Lambda_N = \frac{\Lambda_{P_1} + \Lambda_{P_2}}{\|\overrightarrow{P_1 P_2}\|}$$





"La dernière chose qu'on trouve en faisant  
un ouvrage est de savoir celle qu'il faut  
mettre la première."

Blaise Pascal.



A U T O R I S A T I O N de S O U T E N A N C E

VU les dispositions de l'article 15 Titre III de l'arrêté du 5 juillet 1984 relatif aux études doctorales

VU les rapports de présentation de Messieurs

. N. AYACHE , Directeur de Recherche

. A. LUX , Maître de Conférences

**Monsieur RAMPARANY Solofoarivelo-Fanomezana**

est autorisé(e) à présenter une thèse en soutenance en vue de l'obtention du diplôme  
de DOCTEUR de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, spécialité  
" Informatique "

Fait à Grenoble, le 7 février 1989

*Pour le Président de l'I.N.P.-G.  
et par délégation,  
le Vice-Président  
P. VENNEREAU*

