



**HAL**  
open science

## Contributions à la vision par ordinateur tridimensionnelle

Paul-Louis Borianne

► **To cite this version:**

Paul-Louis Borianne. Contributions à la vision par ordinateur tridimensionnelle. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1984. Français. NNT: . tel-00311470

**HAL Id: tel-00311470**

**<https://theses.hal.science/tel-00311470>**

Submitted on 18 Aug 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 à*

**l'Institut National Polytechnique de Grenoble**

*pour obtenir le grade de*

**DOCTEUR DE 3ème CYCLE**

*par*

**Paul-Louis BORIANNE**



## **CONTRIBUTIONS A LA VISION PAR ORDINATEUR TRIDIMENSIONNELLE**



Thèse soutenue le 10 avril 1984 devant la Commission d'Examen :

Monsieur	G. VEILLON	: Président
Messieurs	J.C. LATOMBE	} Examineurs
	R. MOHR	
	J. MONGE	
	M. NOUGARET	



# THESE

*présentée à*

**l'Institut National Polytechnique de Grenoble**

*pour obtenir le grade de*

**DOCTEUR DE 3ème CYCLE**

*par*

**Paul-Louis BORIANNE**



## **CONTRIBUTIONS A LA VISION PAR ORDINATEUR TRIDIMENSIONNELLE**



Thèse soutenue le 10 avril 1984 devant la Commission d'Examen :

Monsieur	G. VEILLON	: Président
Messieurs	J.C. LATOMBE	} Examineurs
	R. MOHR	
	J. MONGE	
	M. NOUGARET	



**INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE**

**Année universitaire 1982-1983**

**Président de l'Université : D. BLOCH**

**Vice-Président : René CARRE  
Hervé CHERADAME  
Marcel IVANES**

**PROFESSEURS DES UNIVERSITES :**

ANCEAU François	E.N.S.I.M.A.G.
BARRAUD Alain	E.N.S.I.E.G.
BAUDELET Bernard	E.N.S.I.E.G.
BESSON Jean	E.N.S.E.E.G.
BLIMAN Samuel	E.N.S.E.R.G.
BLOCH Daniel	E.N.S.I.E.G.
BOIS Philippe	E.N.S.H.G.
BONNETAIN Lucien	E.N.S.E.E.G.
BONNIER Etienne	E.N.S.E.E.G.
BOUVARD Maurice	E.N.S.H.G.
BRISSONNEAU Pierre	E.N.S.I.E.G.
BUYLE BODIN Maurice	E.N.S.E.R.G.
CAVAIGNAC Jean-François	E.N.S.I.E.G.
CHARTIER Germain	E.N.S.I.E.G.
CHENEVIER Pierre	E.N.S.E.R.G.
CHERADAME Hervé	U.E.R.M.C.P.P.
CHERUY Arlette	E.N.S.I.E.G.
CHIAVERINA Jean	U.E.R.M.C.P.P.
COHEN Joseph	E.N.S.E.R.G.
COUMES André	E.N.S.E.R.G.
DURAND Francis	E.N.S.E.E.G.
DURAND Jean-Louis	E.N.S.I.E.G.
FELICI Noël	E.N.S.I.E.G.
FOULARD Claude	E.N.S.I.E.G.
GENTIL Pierre	E.N.S.E.R.G.
GUERIN Bernard	E.N.S.E.R.G.
GUYOT Pierre	E.N.S.E.E.G.
IVANES Marcel	E.N.S.I.E.G.
JAUSSAUD Pierre	E.N.S.I.E.G.
JOUBERT Jean-Claude	E.N.S.I.E.G.
JOURDAIN Geneviève	E.N.S.I.E.G.
LACOUME Jean-Louis	E.N.S.I.E.G.
LATOMBE Jean-Claude	E.N.S.I.M.A.G.

.../...

LESSIEUR Marcel	E.N.S.H.G.
LESPINARD Georges	E.N.S.H.G.
LONGUEUE Jean-Pierre	E.N.S.I.E.G.
MAZARE Guy	E.N.S.I.M.A.G.
MOREAU René	E.N.S.H.G.
MORET Roger	E.N.S.I.E.G.
MOSSIERE Jacques	E.N.S.I.M.A.G.
PARIAUD Jean-Charles	E.N.S.E.E.G.
PAUTHENET René	E.N.S.I.E.G.
PERRET René	E.N.S.I.E.G.
PERRET Robert	E.N.S.I.E.G.
PIAU Jean-Michel	E.N.S.H.G.
POLOJADOFF Michel	E.N.S.I.E.G.
POUPOT Christian	E.N.S.E.R.G.
RAMEAU Jean-Jacques	E.N.S.E.E.G.
RENAUD Maurice	U.E.R.M.C.P.P.
ROBERT André	U.E.R.M.C.P.P.
ROBERT François	E.N.S.I.M.A.G.
SABONNADIÈRE Jean-Claude	E.N.S.I.E.G.
SAUCIER Gabrielle	E.N.S.I.M.A.G.
SCHLENKER Claire	E.N.S.I.E.G.
SCHLENKER Michel	E.N.S.I.E.G.
SERMET Pierre	E.N.S.E.R.G.
SILVY Jacques	U.E.R.M.C.P.P.
SOHM Jean-Claude	E.N.S.E.E.G.
SOUQUET Jean-Louis	E.N.S.E.E.G.
VEILLON Gérard	E.N.S.I.M.A.G.
ZADWORNY François	E.N.S.E.R.G.

#### PROFESSEURS ASSOCIES

BASTIN Georges	E.N.S.H.G.
BERRIL John	E.N.S.H.G.
CARREAU Pierre	E.N.S.H.G.
GANDINI Alessandro	U.E.R.M.C.P.P.
HAYASHI Hirashi	E.N.S.I.E.G.

#### PROFESSEURS UNIVERSITE DES SCIENCES SOCIALES (Grenoble II)

BOLLIET Louis  
Chatelin Françoise

#### PROFESSEURS E.N.S. Mines de Saint-Etienne

RIEU Jean  
SOUSTELLE Michel

#### CHERCHEURS DU C.N.R.S.

FRUCHART Robert  
VACHAUD Georges

Directeur de Recherche  
Directeur de Recherche

.../...

ALLIBERT Michel	Maître de Recherche
ANSARA Ibrahim	Maître de Recherche
ARMAND Michel	Maître de Recherche
BINDER Gilbert	
CARRE René	Maître de Recherche
DAVID René	Maître de Recherche
DEPORTES Jacques	
DRIOLE Jean	Maître de Recherche
GIGNOUX Damien	
GIVORD Dominique	
GUELIN Pierre	
HOPFINGER Emil	Maître de Recherche
JOUD Jean-Charles	Maître de Recherche
KAMARINOS Georges	Maître de Recherche
KLEITZ Michel	Maître de Recherche
LANDAU Ioan-Dore	Maître de Recherche
LASJAUNIAS J.C.	
MERMET Jean	Maître de Recherche
MUNIER Jacques	Maître de Recherche
PIAU Monique	
PORTESEIL Jean-Louis	
THOLENCE Jean-Louis	
VERDILLON André	

**CHERCHEURS du MINISTERE de la RECHERCHE et de la TECHNOLOGIE (Directeurs et Maîtres de Recherches, ENS Mines de St. Etienne)**

LESBATS Pierre	Directeur de Recherche
BISCONDI Michel	Maître de Recherche
KOBYLANSKI André	Maître de Recherche
LE COZE Jean	Maître de Recherche
LALAUZE René	Maître de Recherche
LANCELOT Francis	Maître de Recherche
THEVENOT François	Maître de Recherche
TRAN MINH Canh	Maître de Recherche

**PERSONNALITES HABILITEES à DIRIGER des TRAVAUX de RECHERCHE (Décision du Conseil Scientifique)**

ALLIBERT Colette	E.N.S.E.E.G.
BERNARD Claude	E.N.S.E.E.G.
BONNET Rolland	E.N.S.E.E.G.
CAILLET Marcel	E.N.S.E.E.G.
CHATILLON Catherine	E.N.S.E.E.G.
CHATILLON Christian	E.N.S.E.E.G.
COULON Michel	E.N.S.E.E.G.
DIARD Jean-Paul	E.N.S.E.E.G.
EUSTAPOPOULOS Nicolas	E.N.S.E.E.G.
FOSTER Panayotis	E.N.S.E.E.G.

.../...

GALERIE Alain	E.N.S.E.E.G.
HAMMOU Abdelkader	E.N.S.E.E.G.
MALMEJAC Yves	E.N.S.E.E.G. (CENG)
MARTIN GARIN Régina	E.N.S.E.E.G.
NGUYEN TRUONG Bernadette	E.N.S.E.E.G.
RAVAINE Denis	E.N.S.E.E.G.
SAINFORT	E.N.S.E.E.G. (CENG)
SARRAZIN Pierre	E.N.S.E.E.G.
SIMON Jean-Paul	E.N.S.E.E.G.
TOUZAIN Philippe	E.N.S.E.E.G.
URBAIN Georges	E.N.S.E.E.G. (Laboratoire des ultra-réfractaires ODEILLON)
GUILHOT Bernard	E.N.S. Mines Saint Etienne
THOMAS Gérard	E.N.S. Mines Saint Etienne
DRIVER Julien	E.N.S. Mines Saint Etienne
BARIBAUD Michel	E.N.S.E.R.G.
BOREL Joseph	E.N.S.E.R.G.
CHOVET Alain	E.N.S.E.R.G.
CHEHIKIAN Alain	E.N.S.E.R.G.
DOLMAZON Jean-Marc	E.N.S.E.R.G.
HERAULT Jeanny	E.N.S.E.R.G.
MONLLOR Christian	E.N.S.E.R.G.
BORNARD Guy	E.N.S.I.E.G.
DESCHIZEAU Pierre	E.N.S.I.E.G.
GLANGEAUD François	E.N.S.I.E.G.
KOFMAN Walter	E.N.S.I.E.G.
LEJEUNE Gérard	E.N.S.I.E.G.
MAZUER Jean	E.N.S.I.E.G.
PERARD Jacques	E.N.S.I.E.G.
REINISCH Raymond	E.N.S.I.E.G.
ALEMANY Antoine	E.N.S.H.G.
BOIS Daniel	E.N.S.H.G.
DARVE Félix	E.N.S.H.G.
MICHEL Jean-Marie	E.N.S.H.G.
OBLED Charles	E.N.S.H.G.
ROWE Alain	E.N.S.H.G.
VAUCLIN Michel	E.N.S.H.G.
WACK Bernard	E.N.S.H.G.
BERT Didier	E.N.S.I.M.A.G.
CALMET Jacques	E.N.S.I.M.A.G.
COURTIN Jacques	E.N.S.I.M.A.G.
COURTOIS Bernard	E.N.S.I.M.A.G.
DELLA DORA Jean	E.N.S.I.M.A.G.
FONLUPT Jean	E.N.S.I.M.A.G.
SIFAKIS Joseph	E.N.S.I.M.A.G.
CHARUEL Robert	U.E.R.M.C.P.P.
CADET Jean	C.E.N.G.
COEURE Philippe	C.E.N.G. (LETI)

.../...

DELHAYE Jean-Marc	C.E.N.G. (STT)
DUPUY Michel	C.E.N.G. (LETI)
JOUBE Hubert	C.E.N.G. (LETI)
NICOLAU Yvan	C.E.N.G. (LETI)
NIFENECKER Hervé	C.E.N.G.
PERROUD Paul	C.E.N.G.
PEUZIN Jean-Claude	C.E.N.G. (LETI)
TAIEB Maurice	C.E.N.G.
VINCENDON Marc	C.E.N.G.

#### **LABORATOIRES EXTERIEURS**

DEMOULIN Eric	C.N.E.T.
DEVINE	C.N.E.T. (R.A.B.)
GERBER Roland	C.N.E.T.
MERCKEL Gérard	C.N.E.T.
PAULEAU Yves	C.N.E.T.
GAUBERT C.	I.N.S.A. Lyon



Je tiens à dédier cette thèse à mon père, qui a su  
me convaincre du plaisir que constitue tout travail de  
recherche, et de la nécessité de toujours se dépasser.



Je remercie chaleureusement toutes les personnes qui m'ont aidé dans mon travail de recherche, d'une manière ou d'une autre, en particulier Jean-Claude Latombe, qui a guidé mon investigation et qui a su modifier et trier avec esprit de synthèse les textes qui ont servi de base à ce rapport de thèse, l'ensemble de l'équipe "Intelligence Artificielle et Robotique", qui m'a fourni un cadre conceptuel pour le développement de mes recherches, à l'occasion de nombreuses heures de discussion ou de séminaires, l'atelier "micro-informatique" et le service d'électronique de l'IMAG, Pascal Di Giacomo et Emmanuel Mazer qui m'ont aidé pour toute la partie matérielle et micro-programmée du capteur, Gérard Mézin avec qui j'ai collaboré pour la réalisation du matériel et du logiciel de base du capteur et qui n'a pas ménagé ses efforts, Jean-François Miribel à qui je dois l'interpréteur de ce logiciel et Yves Demazeau pour les corrections qu'il a apportées à mon travail.

Je remercie aussi toutes les personnes n'ayant pas un rapport direct avec ces recherches, mais qui m'ont permis par leur compréhension et leur sympathie de mener à bien ce travail.

Je tiens enfin à remercier Messieurs Mohr, Monge et Nougaret, membres du Jury, ainsi que Monsieur Veillon qui le préside.



SOMMAIRE

INTRODUCTION

CHAPITRE 1

PRESENTATION DU CAPTEUR

1. PRINCIPE
2. REALISATION MATERIELLE
3. COMPARAISON AVEC D'AUTRES CAPTEURS

CHAPITRE 2

LOGICIEL DE BASE DU CAPTEUR

1. LE "QUOI"
  - 1.1 Calibrage du capteur
  - 1.2 Construction d'image 3D
  - 1.3 Visualisation d'image 3D
2. LE "COMMENT"
  - 2.1 Calibrage du capteur
  - 2.2 Construction d'image 3D
3. QUELQUES ELEMENTS DE PERFORMANCE
  - 3.1 Vitesse
  - 3.2 Précision

CHAPITRE 3

QUELQUES APPLICATIONS DU CAPTEUR

1. DETECTION DE DEFAUTS VOLUMIQUES
2. SUIVI DE JOINTS DE SOUDURE
3. INSERTION DE COMPOSANTS NON STANDARD

## CHAPITRE 4

## SEGMENTATION D'IMAGE 3D

## 1. QUELQUES METHODES DE SEGMENTATION

## 1.1 Transformée de Hough

## 1.1.1 Principe

## 1.1.2 Application aux images 3D

## 1.2 Croissance de région

## 1.2.1 Principe

## 1.2.2 Application aux images 3D

## 1.3 Découpage récursif

## 1.4 Analyse de coupes

## 2. NOTRE METHODE

## 2.1 Cas des objets polyédriques

## 2.1.1 Détection locale des plans

## 2.1.2 Seuillage adaptatif

## 2.1.3 Analyse de connexité

## 2.1.4 Calcul des équations de plans

## 2.1.5 Croissance de région

## 2.1.6 Détermination des arêtes occultantes

## 2.1.7 Autre exemple

## 2.2 Cas des quadriques

## 3. DISCUSSION

## CHAPITRE 5

## CONSTRUCTION AUTOMATIQUE DE

## MODELES D'OBJETS

## 1. PRINCIPE DE LA MISE EN CORRESPONDANCE ENTRE DEUX MODELES PARTIELS

## 2. GENERATION D'HYPOTHESES

## 3. FUSION DE DEUX MODELES

4. CONSTRUCTION D'UN MODELE D'OBJET
5. APPLICATION A L'IDENTIFICATION/LOCALISATION D'OBJETS

## CHAPITRE 6

## INTERACTION 2D-3D

1. INTERET DE L'INTERACTION 2D-3D
  - 1.1 Complémentarité des informations 2D et 3D
  - 1.2 Intégration de la forme et de l'aspect visuel
  - 1.3 Aide à la modélisation
2. CALIBRAGE 2D-3D
  - 2.1 Principe
    - 2.1.1 Sens 3D--->2D
    - 2.1.2 Sens 2D--->3D
  - 2.2 Logiciel de base
    - 2.2.1 Le "quoi"
      - 2.2.1.1 Calibrage espace <---> caméra
      - 2.2.1.2 Primitives diverses
    - 2.2.2 Le "comment"
      - 2.2.2.1 Calibrage espace <---> caméra
      - 2.2.2.2 Primitives diverses

CONCLUSION

BIBLIOGRAPHIE

## ANNEXE A

## DESCRIPTION MATERIELLE DU CAPTEUR

1. L'ensemble caméra-laser et son électronique de prétraitement
2. Le plateau translatable
3. Le calculateur de traitement
4. Le système graphique

## ANNEXE B

## LES FONCTIONS DU LOGICIEL DE BASE

1. Modes d'accès et contenu
2. Commande du plateau translatable seul
3. Prise d'images et visualisation sans déplacement
4. Prise d'image et déplacement combinés

## ANNEXE C

## LOGICIEL DE MOUVEMENT DU PLATEAU

1. Matériel utilisé pour l'échange et principe
2. Protocole d'échange
3. Le logiciel implanté sur le 8080
4. Le logiciel implanté sur le LSI 11/23

## ANNEXE D

## MODELISATION MATHEMATIQUE D'UNE CAMERA

## APPLICATION AU CAPTEUR 3D

1. Spécification géométrique et fonctionnelle d'une caméra
2. Calibrage d'une caméra fixe
3. Relation entre les spécifications géométrique et fonctionnelle d'une caméra
  - 3.1 Géométrique ---> fonctionnel
  - 3.2 Fonctionnel ---> géométrique
4. Calibrage d'une caméra mobile
5. Applications
  - 5.1 Calibrage plan
  - 5.2 Capteur 3D à plan lumineux mobile
  - 5.3 Vision stéréoscopique
6. Calcul des points de fuite

7. Démonstrations

7.1 Démonstration 1

7.2 Démonstration 2

7.3 Démonstration 3

ANNEXE E

CALCUL OPTIMISE DU MEILLEUR PLAN

EN UN POINT D'UNE IMAGE 3D



L'importance de la perception visuelle dans un domaine tel que la Robotique n'est plus à démontrer. En effet, l'information visuelle est particulièrement riche, et la maîtrise de son traitement depuis l'acquisition jusqu'à l'interprétation permet de résoudre des tâches essentielles d'identification et localisation d'objets, d'inspection visuelle, de contrôle de mouvement, etc...

La vision artificielle pour la Robotique reste toutefois limitée à l'analyse d'images en intensité lumineuse, souvent binarisées. Ces images ne permettent d'obtenir assez directement que des informations bidimensionnelles (2D) sur les scènes. Ainsi sont-elles fréquemment utilisées pour localiser des objets posés sur un plan de travail.

L'expérience a montré que l'acquisition de données tridimensionnelles (3D) sur l'environnement d'un robot est souvent indispensable: suivi de joints, identification d'objets se recouvrant partiellement, détermination de relations spatiales entre objets, etc... Cette acquisition peut être réalisée à l'aide de capteurs tactiles, mais ceux-ci nécessitent des actions sur l'environnement et ne fournissent en un temps raisonnable que des informations sur une zone limitée de l'environnement. D'où l'intérêt croissant pour la vision artificielle 3D au cours de ces dernières années.

Contrairement à la vision 2D qui possède un capteur privilégié, la caméra, la vision 3D n'en a pas. Certes on peut obtenir des informations 3D sur une scène à l'aide d'une ou plusieurs caméras (interprétation des ombres et de la texture, stéréovision,...), mais cela nécessite de mettre en oeuvre des méthodes pas toujours bien maîtrisées et demandant des calculs importants. Dans notre travail, nous nous sommes donc intéressés à un type de capteur, combinant caméra et laser, qui permet d'obtenir l'information 3D plus directement. Nous avons commencé l'étude de ce capteur en stage de DEA [Borianne-81].

Notre mémoire comporte six chapitres.

- Le premier chapitre décrit le capteur 3D que nous avons réalisé, constitué d'une caméra et d'un plan lumineux émis par un laser disposant d'une lentille cylindrique. L'intersection du plan lumineux avec la droite de visée d'un pixel  $P: (i, j)$  détermine un point  $M$  dans l'espace.
- Le second chapitre décrit l'environnement logiciel de ce capteur, qui inclut notamment une procédure de calibrage, ainsi que des primitives de construction et de visualisation d'images 3D.
- Le troisième chapitre montre trois applications du capteur: détection de défauts volumiques, suivi de joints de soudure et insertion de composants électroniques non standard sur une carte de circuit imprimé.
- Le chapitre 4 présente un procédé de segmentation d'images 3D en primitives telles que plans et quadriques. Une partie des calculs peut se faire en mode "pipe-line", et une certaine fiabilité dans les résultats est obtenue par l'utilisation de deux niveaux: local et global.

- Le chapitre 5 traite du problème de la correspondance de modèles 3D, avec comme application la reconstruction de modèles complets à partir d'une liste de modèles partiels.
- Le chapitre 6 aborde le problème de l'utilisation conjointe d'un capteur 3D et d'un capteur 2D, avec comme application l'obtention d'images 3D incluant des données sur la nature visuelle des surfaces de l'objet.

CHAPITRE 1

PRESENTATION DU CAPTEUR

1. Principe
2. Réalisation matérielle
3. Comparaison avec d'autres capteurs



Nous présentons d'abord le principe fonctionnel du capteur que nous avons mis en oeuvre (paragraphe 1). Ensuite nous décrivons sa réalisation matérielle (paragraphe 2). Enfin nous le comparons à d'autres capteurs 3D (paragraphe 3).

## 1. PRINCIPE

Le capteur de vision 3D que nous avons mis en oeuvre se compose de trois parties illustrées à la figure 1.1:

- une source de lumière,
- une caméra vidéo,
- un plateau translatable.

La fonction de la source de lumière est d'émettre un plan de lumière qui coupe les objets de la scène à analyser suivant un trait lumineux, appelé le trait lumineux réel.

La caméra a pour fonction de fournir une image du trait lumineux. Son axe de vision coupe le plan lumineux suivant un angle  $\theta$ . Elle est munie d'un numériseur qui construit une image digitalisée  $I(i,j)$  contenant le trait lumineux vu. Elle est solidaire de la source de lumière.

A chaque pixel  $P$  de l'image  $I(i,j)$  correspond un point  $Q$  sur le plan rétinien (surface photosensible) de la caméra. Le pixel  $P$  représente donc ce que voit la caméra suivant la droite passant par  $Q$  et le point focal de l'objectif (droite de visée de  $P$ ). Cette droite a une équation dans un repère cartésien attaché à l'ensemble caméra-laser qui est parfaitement déterminée par la géométrie du capteur, l'optique de la caméra et le numériseur.

Chaque pixel  $P$  de  $I(i,j)$  sur le trait lumineux vu correspond donc à un point  $M$  du trait lumineux réel.  $M$  est l'intersection de la droite de visée de  $P$  et du plan lumineux (principe de triangulation, voir figure 1.2). Les coordonnées de  $M$  dans le repère attaché à l'ensemble caméra-laser sont ainsi déterminées.

Il existe donc une transformation  $T'$  qui permet de passer des coordonnées  $i,j$  dans  $I$  aux coordonnées  $X',Y',Z'$  des points du plan lumineux dans un repère attaché au capteur. La détermination des paramètres de  $T'$  constitue le calibrage du capteur (cf. Chapitre 2). Nous supposons dans ce qui suit qu'ils sont connus.

Si l'on connaît la transformation  $T''$  faisant passer du repère  $OX'Y'Z'$  à un repère  $OXYZ$  attaché à la scène, alors  $T=T''T'$  permet de transformer un trait lumineux vu en une liste de triplets  $(X_i, Y_i, Z_i)$  qui représente le trait lumineux réel dans le repère  $OXYZ$ . Le plateau translatable permet de faire défiler la scène devant le capteur, c'est-à-dire relativement au repère attaché au capteur. Pour chaque position du plateau, on obtient une nouvelle liste de coordonnées.

En faisant en sorte que les axes  $OX$ ,  $OY$  et  $OZ$  soient parallèles aux axes  $OX'$ ,  $OY'$  et  $OZ'$  respectivement, et que l'axe de translation du plateau soit parallèle à l'axe  $OZ$ , on réduit  $T''$  à une simple translation dont le seul paramètre variable est la composante

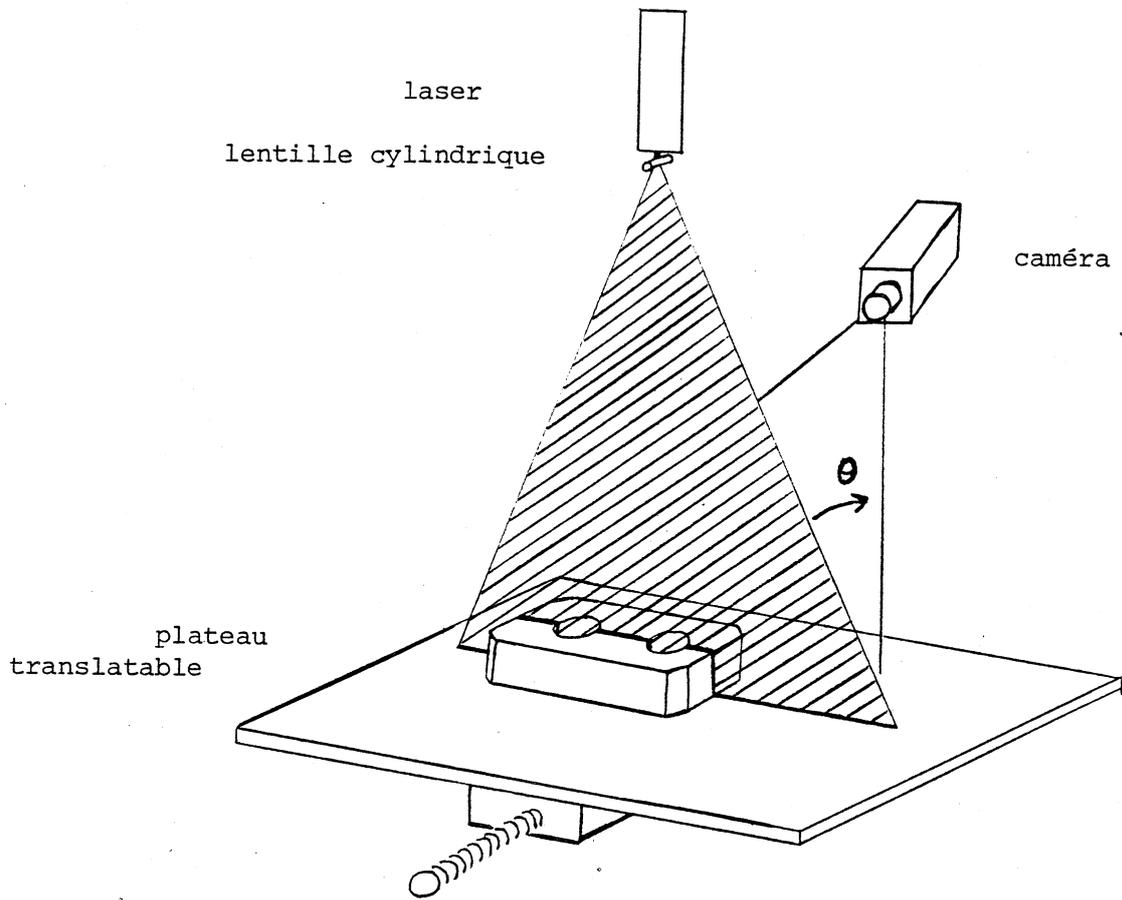


figure 1.1  
Le capteur.

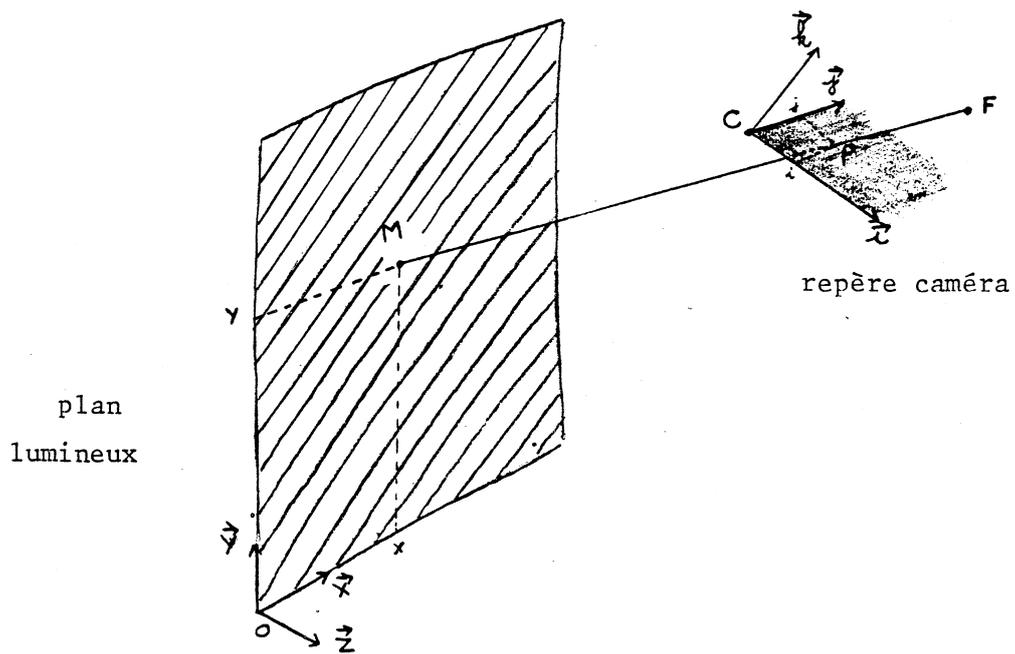


figure 1.2  
Les repères attachés au plan lumineux et  
à la caméra.

suivant  $Z$ . Chaque position relative du plateau et du capteur correspond alors à une valeur  $Z_i$  de  $Z$ , et permet d'obtenir une courbe discrète  $Y_i=f_i(X_i)$  représentant le trait lumineux réel à l'abscisse  $Z_i$  de la scène. Le déplacement du plateau permet de construire ligne par ligne une image discrète  $Y=f(X,Z)$  dans le repère  $OXYZ$ ; c'est l'image 3D de la scène.

Pour certaines valeurs du couple  $(X,Z)$ , la valeur de  $f$  peut ne pas être définie. Ces valeurs correspondent aux points qui ne peuvent pas être vus par la caméra et éclairés par le laser simultanément. Dans la suite, on dira que ces points sont indéfinis. On peut réduire leur nombre en diminuant la valeur de l'angle  $\theta$ , mais alors on réduit aussi la précision du capteur. On pourrait aussi utiliser un capteur combinant de façon judicieuse plusieurs caméras et sources de lumière.

## 2. REALISATION MATERIELLE

La réalisation matérielle du capteur est présentée à la figure 1.3 et illustrée par la photo 1.4.

Dans la version initiale du capteur [Borianne-81], la source de lumière était un projecteur de diapositives, ce qui avait l'inconvénient d'imposer un fonctionnement en "chambre noire". Aujourd'hui elle est constituée par un laser muni d'une lentille cylindrique pour diffracter le rayon en un plan lumineux. La caméra est équipée d'un filtre passe-bande centré sur la longueur d'onde de la lumière laser qui permet de travailler en lumière ambiante.

La caméra est une caméra classique à tube Vidicon ou à état solide (cellule CCD ou CID). Son optique est interchangeable pour l'adapter à l'application considérée. De même la position de la caméra relativement au laser (angle  $\theta$ ) peut être modifiée d'une application à une autre. Toute modification de caméra, d'optique, ou de géométrie de l'ensemble caméra-laser requiert bien sûr un recalibrage du capteur. Mais comme nous le verrons au chapitre 2, l'opération de calibrage s'effectue très simplement, ce qui permet de reconfigurer rapidement le capteur en fonction des problèmes abordés.

La numérisation du signal vidéo de la caméra conduit à un trait lumineux vu qui est épais de plusieurs pixels (typiquement 3 à 8 pixels suivant l'optique, voir photo 1.5 (a)). Afin d'obtenir un plan de référence fixe, ce trait est "squelettisé" de la façon suivante: on calcule d'abord le gradient lumineux en chaque point de l'image, puis on seuille son module et on garde les points correspondant aux maxima locaux de ce module ("zero-crossing" [Hildreth-80]) pour obtenir les deux bords du trait (voir photo 1.5 (b)), et enfin on conserve l'un de ces bords. Pour simplifier les traitements, la caméra est disposée relativement à la lentille du laser de telle façon que les colonnes de l'image digitalisée  $I(i,j)$  correspondent sur le plan rétinien de la caméra à des droites parallèles au plan émis par le laser. Ainsi, on obtient toujours deux bords du trait lumineux vu par ligne de l'image (si le trait coupe la ligne) et seul celui qui correspond à un gradient négatif est conservé (voir photo 1.5 (c)). Pour certains cas de perspective avec des objets concaves, il est possible que le trait lumineux soit

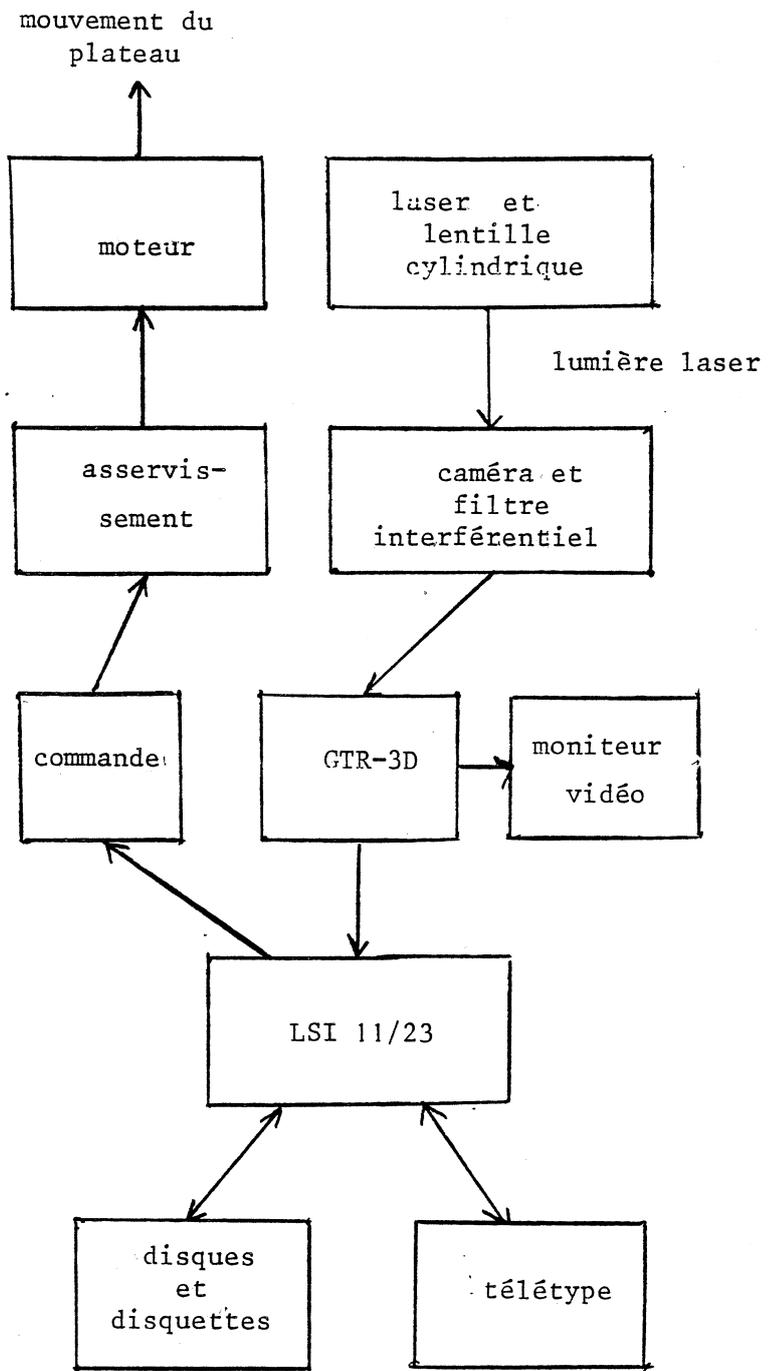


figure 1.3

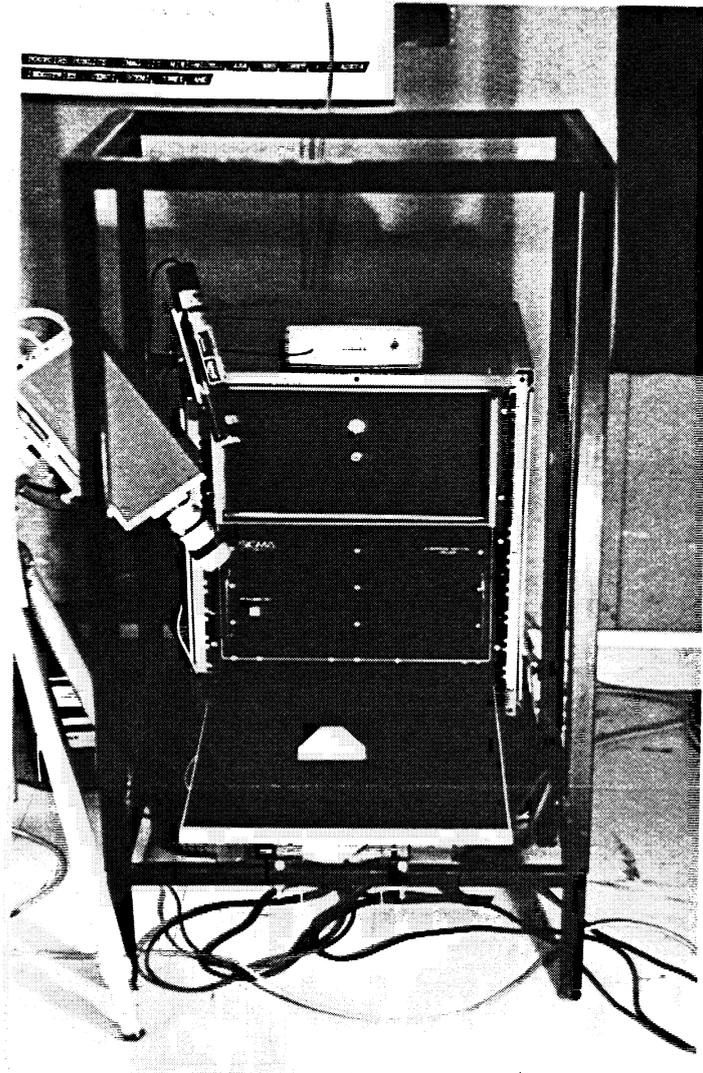


photo 1.4

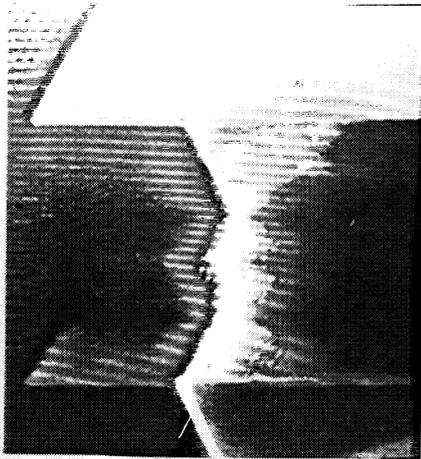


photo 1.5 (a)



photo 1.5 (b)

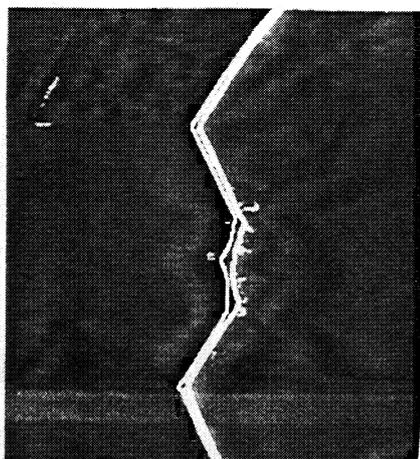


photo 1.5 (c)

vu plusieurs fois dans la même ligne; c'est la première occurrence qui est alors choisie.

La numérisation du signal vidéo et sa squelettisation sont effectuées en temps réel (c'est-à-dire à la vitesse de défilement du signal vidéo) sur une trame (soit en 1/50 ième de seconde) par un processeur en électronique câblée, GTR-3D, conçu au Laboratoire de Traitement d'Images et de Reconnaissance des Formes [Chéhikian-81]. Ce processeur effectue le calcul du gradient par un opérateur de Sobel  $3 \times 3$ . Il transmet au calculateur un tableau CONTOUR [1::256] représentant le trait lumineux vu squelettisé: CONTOUR(j) a pour valeur l'abscisse i ( $i \in [1,256]$ ) du trait lumineux à l'ordonnée j. S'il n'y a pas de point dans la ligne, le processeur retourne la valeur 0.

En plus des opérations de numérisation et de squelettisation, le processeur câblé effectue une "analyse de signature" du trait lumineux par mesure de son épaisseur. Cette analyse permet d'éliminer une grande partie du bruit lorsque le capteur travaille en présence d'une autre source de lumière émettant dans la longueur d'onde du laser. Les photos 1.6 (a)(b) et (c) illustrent ce traitement: l'image (a) est l'image digitalisée; l'image (b) montre les points à fort gradient et l'image (c) les points sélectionnés par GTR-3D comme appartenant au bord droit du trait lumineux (en surluminance sur la photo).

Le plateau translatable est un support horizontal motorisé et asservi en position. L'ensemble caméra-laser est placé au-dessus de ce support (voir photo 1.4).

L'ensemble du logiciel décrit dans la suite de ce rapport est implanté sur un calculateur LSI 11/23 relié à GTR-3D, au système de pilotage du plateau et à un système graphique permettant la visualisation d'images à plusieurs niveaux de gris. Entre autres fonctions, ce logiciel construit les images  $Y=f(X,Z)$  en coordonnant les saisies d'images (obtention des tableaux CONTOUR [1::256] sur le processeur GTR-3D) et les translations du plateau. En choisissant le pas de ces translations, il fixe la résolution des images 3D suivant Z.

Une description des matériels effectivement mis en oeuvre est donnée dans l'annexe A.

### 3. COMPARAISON AVEC D'AUTRES CAPTEURS

Il existe de nombreux systèmes à base de capteurs optiques permettant d'obtenir l'image 3D d'une scène. Nous distinguons ci-dessous les systèmes indirects et les systèmes directs. Les premiers nécessitent des prétraitements importants, alors que les seconds fournissent l'information 3D de façon quasi-immédiate. Cette distinction n'est bien entendue pas absolue.

Parmi les systèmes indirects, le plus connu est la stéréovision passive qui utilise deux caméras (voir par exemple [Mohr-84] [Grimson-81] [Genery-81] et [Mayhew-81]). L'obtention de l'information 3D passe par une corrélation des deux images obtenues, qui établit une correspondance entre les pixels; l'information 3D

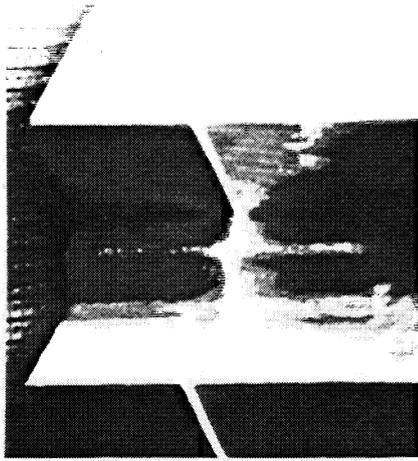


photo 1.6 (a)

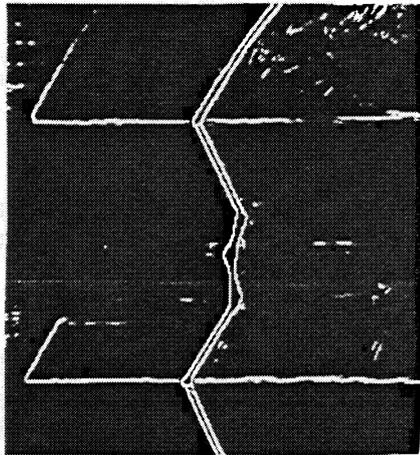


photo 1.6 (b)

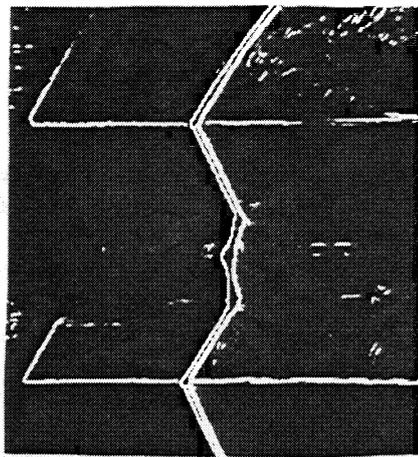


photo 1.6 (c)

s'en déduit par triangulation (intersection des droites de visée de deux pixels en correspondance). D'autres systèmes utilisent une seule caméra; ils reposent alors par exemple sur une méthode de photométrie (calcul des normales aux surfaces à partir de la luminance et de la répartition des niveaux de gris dans l'image) [Binford-81] [Ikeuchi-81] [Woodham-78] ou sur une méthode de gradient de texture (les éléments de texture ont une taille qui diminue avec la distance) [Kender-79] [Ohta-81] [Witkin-81]. Les systèmes monoculaires peuvent aussi reposer sur une utilisation de modèles a priori des objets de la scène [Barrow-81] [Sugihara-77].

D'une façon générale, les systèmes indirects sont lents. De plus, le problème de l'obtention de l'information 3D dans ces systèmes n'est pas parfaitement résolu : ambiguïté en stéréovision, établissement de conditions aux limites avec la méthode de photométrie, ... Enfin les systèmes monoculaires fondés sur la photométrie ou le gradient de texture fournissent des informations 3D relatives, et non pas absolues.

Les systèmes directs sont nombreux. Parmi ceux qui ont été étudiés dans le cadre de la Robotique, on peut citer les interféromètres laser fondés sur la mesure du temps de vol [Duda-79] et les associations caméra-laser utilisant la triangulation [Agin-79] [Bolles-83] [Cain-80] [Faugeras-82] [Gelbgras-84] [Laurgeau-82] [Monge-82] [Popplestone-77] [Shirai-71]. D'autres systèmes directs pourraient être imaginés sur la base de techniques différentes telles que l'Holographie ou le Moiré.

Les interféromètres laser précis ont l'inconvénient d'être chers et assez lents. Il en est de même des associations caméra-laser utilisant des déflecteurs pour orienter le rayon laser [Faugeras-82] [Monge-82]; il faut alors une prise d'image par orientation du rayon. De plus, l'utilisation de déflecteurs peut être délicate dans certaines applications.

D'une façon générale, le choix du principe de notre capteur a été motivé par sa grande simplicité qui permet une mise en oeuvre pratique très peu coûteuse. Il résulte aussi de la relative rapidité qui en découle, puisque chaque ligne d'une image 3D  $Y=f(X,Z)$  peut être obtenue en une seule acquisition d'image 2D. On notera cependant qu'une partie importante des logiciels que nous avons développés, notamment ceux décrits aux chapitres 4 et 5, sont indépendants du principe du capteur.

Au niveau de sa mise en oeuvre, notre capteur se distingue d'autres capteurs fondés sur le même principe [Agin-79] [Bolles-83]

[Laurgeau-82] [Popplestone-77] [Shirai-71] par son prétraitement du trait lumineux vu:

- La plupart des systèmes squelettisent le trait en binarisant l'image, puis en prenant le point milieu, ce qui correspond en général à une référence moins fidèle que le bord obtenu par le calcul de gradient.
- Aucun système n'effectue l'analyse de signature présentée plus haut ce qui les rend plus sensibles aux émissions de lumière parasites dans la longueur d'onde du laser.
- Dans la plupart des systèmes, les prétraitements sont effectués par logiciel, ce qui rend l'acquisition d'une image 3D beaucoup plus longue.

Il se distingue aussi de plusieurs d'entre eux par la possibilité de calibrage automatique, qui permet de le reconfigurer aisément en faisant varier soit les paramètres d'orientation ou de position de la caméra (paramètres externes de la caméra), soit les paramètres de mise au point, grossissement, diaphragme (paramètres internes de la caméra).

Notre capteur n'a qu'un degré de liberté de mouvement relativement à la scène. En permettant au capteur de "tourner" autour de la scène, on aurait pu construire des images 3D "complètes" de la forme  $g(X,Y,Z)=b$ , où  $b \in \{0,1\}$ , de telle façon que  $b=1$  si  $(X,Y,Z)$  est un point d'un objet. De telles images sont plus générales que les images  $Y=f(X,Z)$ , mais ces dernières ont l'avantage d'avoir une représentation matricielle particulièrement simple et compacte. En effet, une image 3D de la forme  $g(X,Y,Z)=b$  nécessite une place mémoire importante. On peut aussi choisir de représenter une image 3D "complète" en dressant une liste de tous les points des objets, mais cette description a le désavantage de ne pas faire apparaître explicitement les relations de connexité entre les points, qu'il faut alors représenter dans une autre structure de donnée [Pauchon-83]. Par ailleurs, le mouvement de translation qui anime les scènes relativement à notre capteur est facile à mettre en oeuvre; il correspond typiquement au mouvement d'un tapis roulant sur une chaîne de fabrication industrielle.

Pour une description plus détaillée des différents capteurs 3D, tant optiques que mécaniques, ou autres, on peut se reporter à [Ballard-82] [Hébert-83] ou à [Lelandais-84].

## CHAPITRE 2

## LOGICIEL DE BASE DU CAPTEUR

## 1. Le "quoi"

## 1.1 Calibrage du capteur

## 1.2 Construction d'image 3D

## 1.3 Visualisation d'image 3D

## 2. Le "comment"

## 2.1 Calibrage du capteur

## 2.2 Construction d'image 3D

## 3. Quelques éléments de performance

## 3.1 Vitesse

## 3.2 Précision



Le logiciel de base du capteur 3D assure trois fonctions principales:

- Calibrage du capteur,
- Construction d'image 3D,
- Visualisation d'image 3D.

Dans le paragraphe 1 de ce chapitre, nous présentons le "QUOI" de ce logiciel, c'est-à-dire ce qu'il est capable de faire, ou plus précisément ce que doit savoir son utilisateur. Dans le paragraphe 2, nous décrivons les méthodes et algorithmes utilisés pour le réaliser, c'est à dire le "COMMENT" du logiciel. Le paragraphe 3 présente quelques éléments de performance du capteur muni de son logiciel de base.

L'accès au logiciel peut se faire à travers un langage de commandes interactif décrit dans l'annexe B, qui facilite la mise en oeuvre du capteur sur de nouvelles applications.

## 1. LE "QUOI"

### 1.1 Calibrage du capteur

Il s'agit de déterminer la transformation  $T$  qui permet de passer des coordonnées  $(\text{CONTOUR}(j), j), j \in [1, 256]$ , fournies par GTR-3D aux coordonnées  $X, Y, Z$  des points correspondants de la scène, exprimées dans un repère attaché à celle-ci.

Au chapitre 1, nous avons considéré  $T$  comme la composition de deux transformations  $T'$  et  $T''$ :

- $T'$  fait passer des coordonnées  $(\text{CONTOUR}(j), j)$  aux coordonnées  $X', Y', Z'$  dans un repère  $O'X'Y'Z'$  attaché au capteur.
- $T''$  fait passer des coordonnées  $X'Y'Z'$  aux coordonnées  $XYZ$ .

Pour simplifier les formules de calibrage, sans nuire à la généralité du capteur, nous choisissons les repères  $O'X'Y'Z'$  et  $OXYZ$  de la façon suivante (cf. figure 2.1):

- Le repère  $OXYZ$  est attaché au plateau translatable de telle façon que le plan  $OXZ$  contienne la surface du plateau et que l'axe  $OZ$  soit parallèle à l'axe de translation du plateau.
- Le repère  $O'X'Y'Z'$  se déduit de  $OXYZ$  par une translation suivant l'axe  $OZ$  et est tel que le plan  $OX'Y'$  soit confondu avec le plan lumineux (plus exactement le "bord" du plan lumineux qui correspond au trait squelettisé).

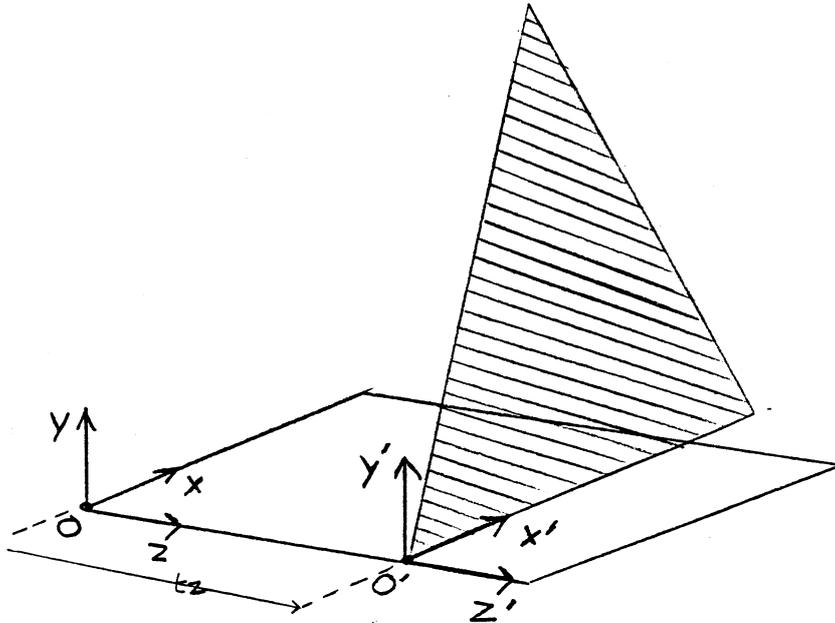


figure 2.1

Ces choix imposent une contrainte sur le montage du capteur relativement au plateau: il faut que le plan lumineux soit perpendiculaire au plateau et à l'axe de translation du plateau.

Une partie du calibrage consiste alors à initialiser la position du plateau relativement au capteur. Cette initialisation se fait en commandant un déplacement du plateau dans un sens imposé jusqu'à ce qu'il soit positionné sur une butée (détectée grâce à un capteur inductif) fixée au capteur (le logiciel permettant de déplacer le plateau est décrit en annexe C). Par définition cette position relative du plateau et du capteur correspond à la superposition de  $O$  et de  $O'$ . Ainsi toute translation ultérieure du plateau de  $-tz$  suivant  $OZ$  correspond à placer le capteur relativement à la scène de telle façon que l'équation du plan lumineux dans  $OXYZ$  soit  $Z=tz$ .

L'autre partie du calibrage consiste à déterminer les paramètres de la transformation  $T'$ , qui, en fonction des repères choisis, fait passer des coordonnées  $(i,j)$  aux coordonnées  $(X',Y')=(X,Y)$ .

Les formules de la transformation étant (cf. annexe D):

$$\begin{cases} X'=X = \frac{b11i + b12j + b14}{b31i + b32j + b34} \\ Y'=Y = \frac{b21i + b22j + b24}{b31i + b32j + b34} \end{cases}$$

il est nécessaire de mettre en correspondance quatre points de l'espace, de coordonnées  $X,Y$  connues, avec quatre points de l'image perçue. En effet, cela nous donne 8 équations qui nous permettent de

calculer les 8 inconnues  $b_{11}, b_{12}, b_{14}, b_{21}, b_{22}, b_{24}, b_{31}, b_{32}$ , une valeur arbitraire (1 par exemple) pouvant être donnée à  $b_{34}$ . Pour obtenir les quatre points de l'espace on utilise des objets étalons tels que ceux de la figure 2.2. Comme le montre la figure un tel objet doit être posé sur le plateau à un endroit fixé de telle sorte que ses faces latérales soient parallèles au plan lumineux et que celui-ci coupe l'objet. L'intersection du plan avec l'objet détermine alors une courbe constituée de cinq segments de droite (pour le second, il faut le considérer avec le plateau sur lequel il est posé, qui donne deux des segments). La détermination des paramètres  $b_{ij}$  se fait alors en mettant en correspondance les quatre points anguleux de la courbe "espace" avec les quatre points anguleux de la courbe "vue". Les premiers sont donnés au système par l'utilisateur. Les seconds sont obtenus à partir de la courbe "vue" par segmentation (voir paragraphe 2.1).

Remarque :

Il est souhaitable que l'objet étalon soit d'une taille analogue à celle des objets des scènes à analyser. En effet, si par exemple l'objet étalon est petit par rapport aux objets des scènes, on peut s'attendre à une mauvaise précision due aux non-linéarités de la caméra et de son optique. Pour certaines applications, on pourrait envisager un multi-étalonnage, en partitionnant l'espace en un certain nombre de volumes; à chaque volume correspondrait un objet étalon de dimensions appropriées.

### 1.2 Construction d'image 3D

Il s'agit de construire une fonction à deux variables discrètes  $Y=f(X,Z)$  (l'image 3D), en effectuant des coupes successives de la scène par le plan lumineux pour différentes positions du plateau. Chaque position du plateau correspond à une valeur de  $t_z$  telle que l'équation du plan lumineux dans OXYZ soit  $Z=t_z$ .

En entrée de la fonction de construction d'image on donne les paramètres suivants (voir figure 2.3):

- $Z_0$  position de la première courbe (réel).
- $\Delta Z$  écart entre deux courbes (réel).
- $n_c$  nombre de courbes (entier).
  
- $X_0$  position du premier point (réel).
- $\Delta X$  écart entre deux points (réel).
- $n_p$  nombre de points (entier).

Les paramètres  $Z_0$  et  $X_0$  sont des paramètres (en centimètres) qui définissent les coordonnées "origines" de l'image 3D suivant OZ et suivant OX; ils n'ont d'importance que si l'on veut comparer plusieurs images 3D. Les paramètres  $n_c$  et  $n_p$  sont les nombres de points qu'on désire pour l'image suivant les axes Z et X. Les paramètres  $\Delta Z$  et  $\Delta X$  correspondent aux pas d'échantillonnage, exprimés en cm, suivant les axes Z et X. Si ces pas ne sont pas les mêmes, alors la résolution n'est pas la même dans les deux directions.

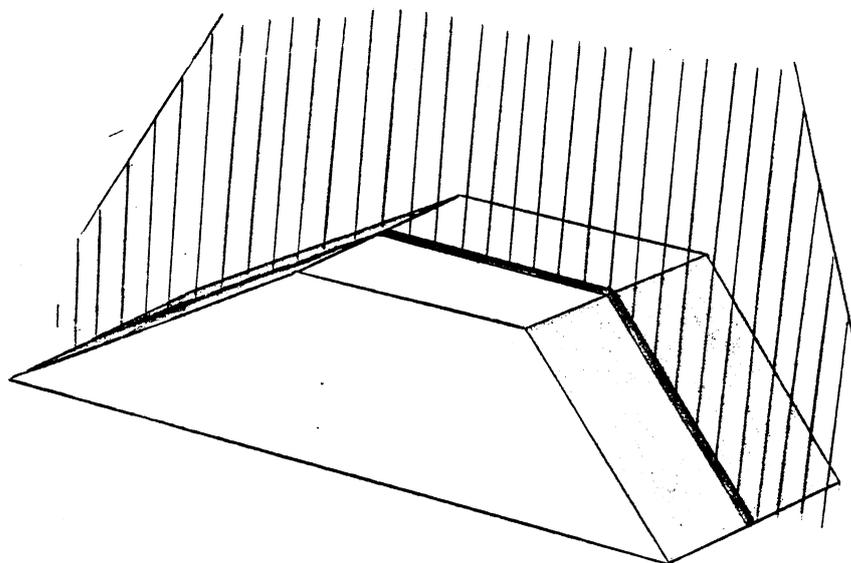
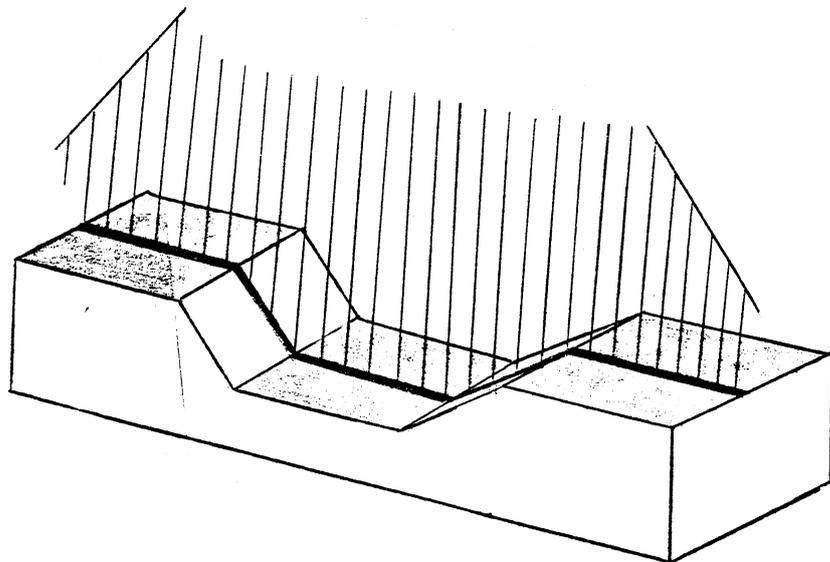


figure 2.2  
Exemples d'objets étalons.

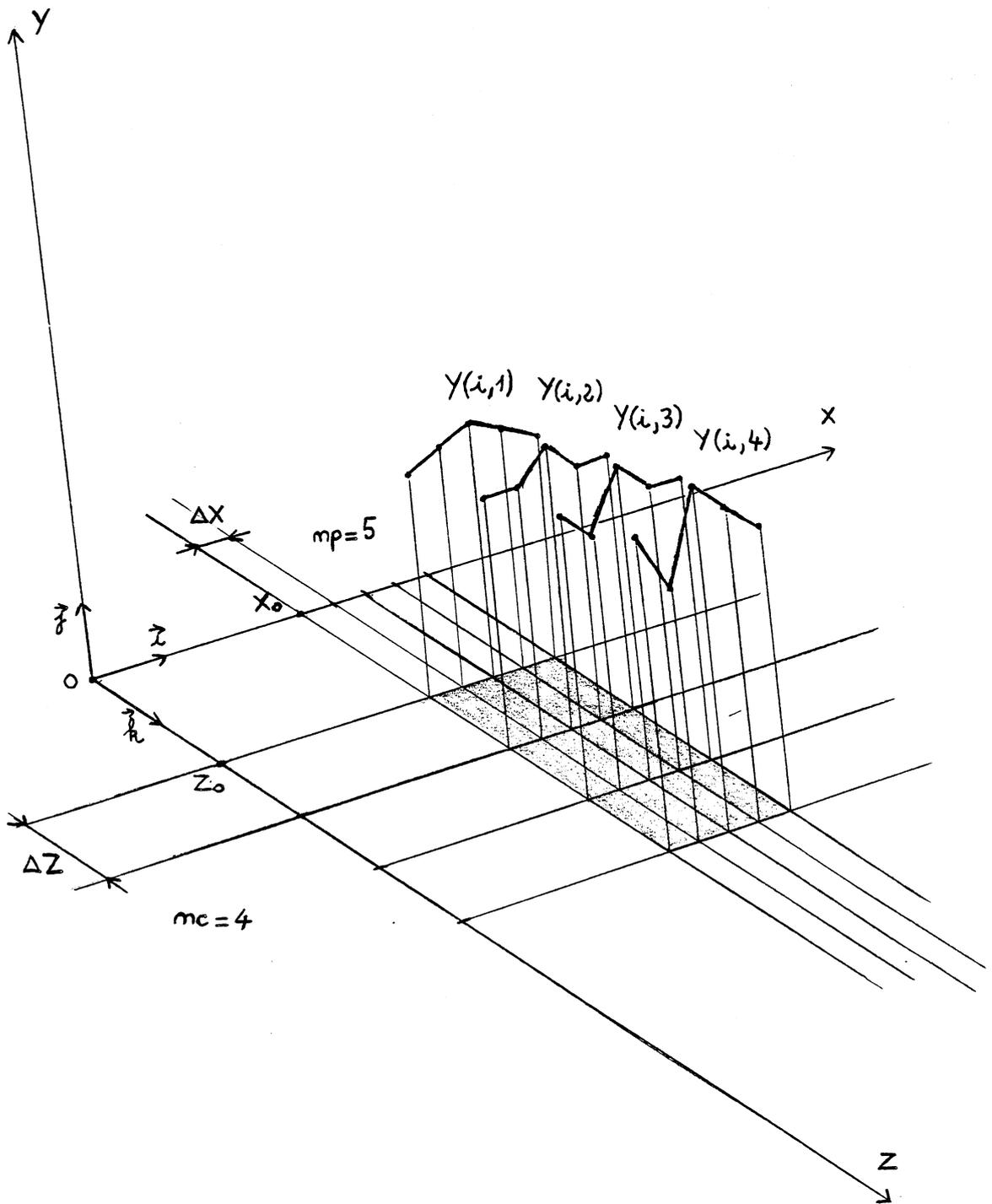


figure 2.3

Les paramètres de la construction d'une image 3D.

L'image  $Y=f(X,Z)$  est mémorisée dans un tableau  $Y(i,j)$ , où  $(i,j) \in [1,np] \times [1,nc]$ , tel que :

$$\begin{cases} X = X_0 + (i-1) \cdot \Delta X \\ Y = Y(i,j) \\ Z = Z_0 + (j-1) \cdot \Delta Z \end{cases}$$

En plus de  $Y(i,j)$ , la prise d'image fournit un autre tableau noté  $D(i,j)$  de même dimension et à valeur dans  $\{0,1\}$ . Ce tableau indique pour chaque point  $(i,j)$ , si la valeur de  $Y(i,j)$  est définie ou non (0 : non défini, 1 : défini). En effet, le capteur peut ne pas assurer une valeur de  $Y(i,j)$  en chaque point, pour les raisons suivantes, liées à la nature de notre capteur :

- Le point n'est pas vu, car il sort du champ de vision de la caméra.
- Le point n'est pas vu, car du point de vue de la caméra, il est caché par une autre partie de l'objet.
- Le point n'est pas éclairé, car du point de vue de la source de lumière, il est caché par une autre partie de l'objet.
- Le point n'est pas vu, car il ne renvoie pas suffisamment de lumière vers la caméra (cela peut être le cas par exemple si le plan lumineux a une incidence trop faible par rapport à l'élément de surface, ou si l'objet est un objet noir mat, qui absorbe trop la lumière).

Remarque 1 :

L'échantillonnage régulier en  $Z$  de l'image découle simplement de la commande du plateau translatable, et va donc de soi. Il n'en est pas de même pour l'autre direction  $X$ . En effet, après squelettisation du trait, on obtient une courbe "écran"  $CONTOUR(j)$ . La transformation  $T$  fait passer de la courbe  $(CONTOUR(j), j)$ ,  $j=1$  à  $256$ , à une courbe  $(X(j), Y(j))$ . Or cette dernière n'est pas nécessairement régulièrement échantillonnée en  $X$ , ce qui est équivalent à dire que les  $X(j)$  ne sont pas régulièrement espacés. Cela est dû essentiellement aux effets de perspective. Il est donc nécessaire de reformater la courbe  $(X(j), Y(j))$  en l'échantillonnant avec un pas régulier.

Remarque 2 :

Pendant toute la prise de vue, le pas d'échantillonnage en  $Z$  est constant. Il n'est pas exclu d'envisager des prises de vues à pas variable, le pas dépendant de la forme de l'objet, la résolution étant plus fine dans les zones de fortes variations. On peut aussi envisager de faire revenir le plateau en arrière pour "affiner" une partie d'image, par exemple après une interprétation partielle. Aucune implantation de ce type n'a été faite ici. Pour plus de détails sur les problèmes de "stratégies de prises de vues" dans le cas des capteurs à plan lumineux, il est intéressant de se reporter

à [Agin-79].

### 1.3 Visualisation d'images 3D

Le logiciel de base permet les visualisations suivantes:

- Représentation graphique de l'image 3D (voir photo 2.4):

Le logiciel permet de représenter, à l'échelle désirée, et dans les bonnes proportions, l'image en superposant les différentes courbes  $Z = Z_0 + (j-1) \Delta Z$  en les espaçant sur l'écran de la distance qui les sépare dans l'image 3D ( $\Delta Z$ ).

- Représentation en intensité lumineuse de l'image 3D (voir photo 2.5):

Le système permet de représenter l'image  $Y=f(X,Z)$  en intensité lumineuse en associant aux niveaux de gris extrêmes (1 et 16 avec le matériel utilisé) les valeurs extrêmes de  $Y$  ( $\text{MIN}(f(X,Z))$  et  $\text{MAX}(f(X,Z))$ ) et en prenant l'écran comme plan OXZ. Ainsi plus un point est haut ( $Y$  grand), plus son intensité lumineuse est grande. Les dimensions d'affichage respectent les proportions de l'image 3D.

#### Remarques :

- Afin d'afficher au mieux l'image 3D en intensité lumineuse, on peut cadrer les niveaux de gris extrêmes (1 et 16), non pas avec les minima et maxima de  $Y$ , ce qui n'est pas toujours satisfaisant, notamment lorsque des points "aberrants" sont possibles, mais avec, pour le niveau 1, la moyenne des  $Y$  moins trois écart-types, et pour le niveau 16, la moyenne des  $Y$  plus trois écart-types. Pour les points qui sont en dehors de l'intervalle, ils sont affichés au niveau 1 ou 16, suivant qu'ils sont plus petits ou plus grands que les bornes.

- Le plateau translatable peut être perçu par le capteur comme faisant partie de la scène. Il est possible de le supprimer si on le désire, afin de ne garder dans les images graphiques ou en intensité lumineuse que les objets de la scène à analyser (voir photo 2.6 (a) et (b)).

## 2. LE "COMMENT"

Dans ce paragraphe nous présentons les méthodes de traitement d'image et de calcul numérique mises en oeuvre pour effectuer le calibrage du capteur et la construction d'images 3D. Les algorithmes de visualisation d'images 3D ne posent pas de problèmes particuliers.

L'ensemble du logiciel de base, à l'exception de quelques fonctions en assembleur, est écrit en FORTRAN sur le calculateur LSI 11/23.

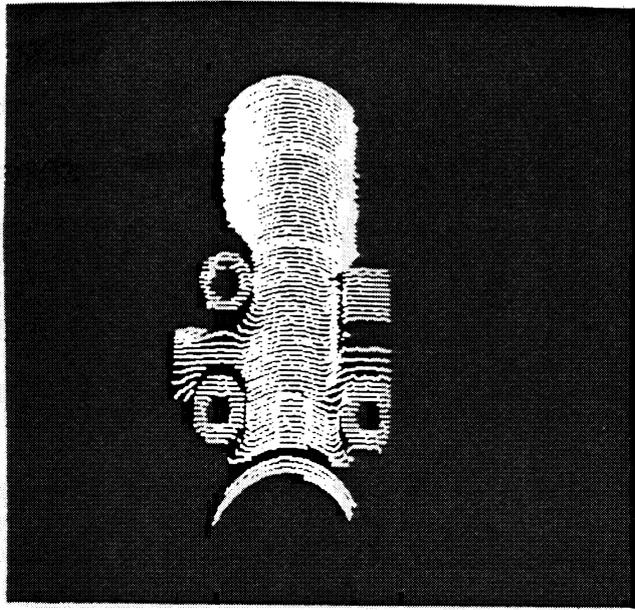


photo 2.4

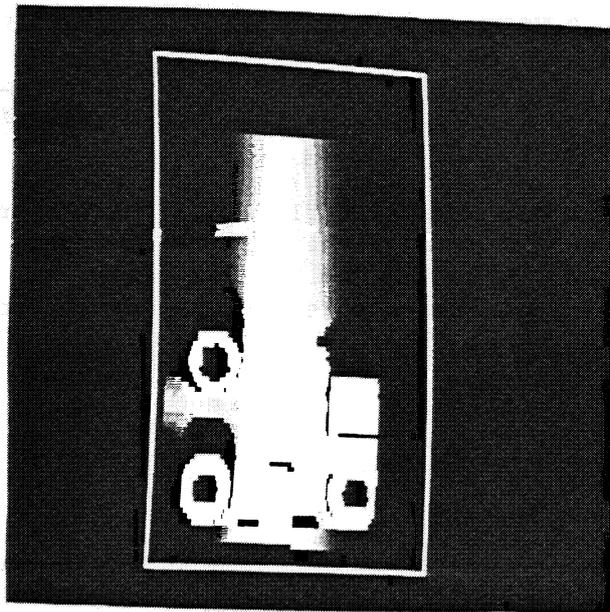


photo 2.5

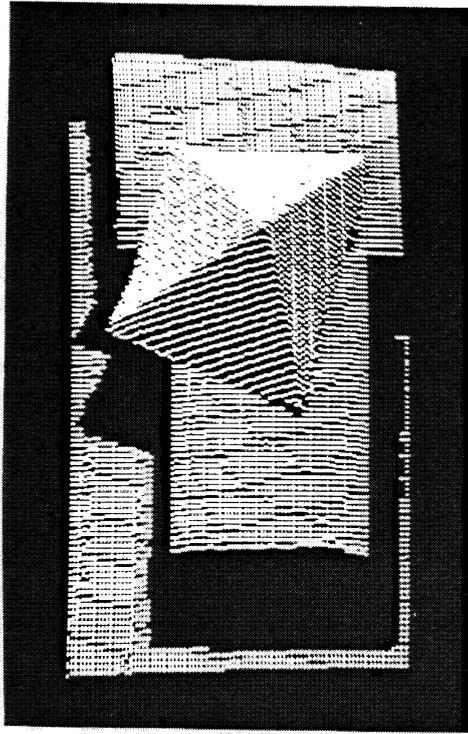


photo 2.6 (a)

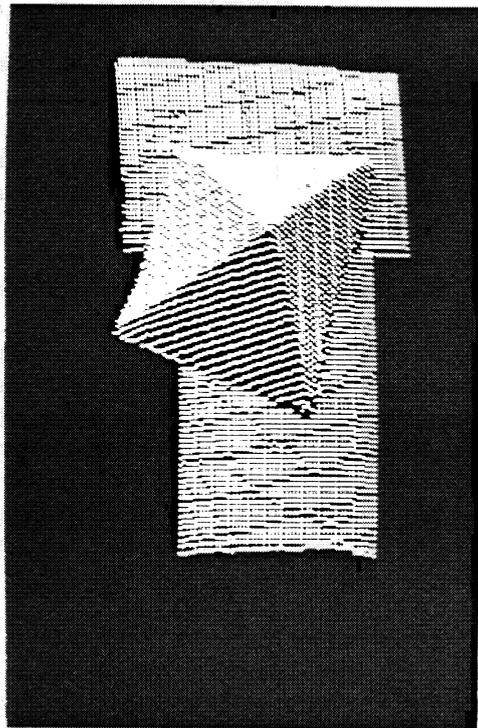


photo 2.6 (b)

## 2.1 Calibrage du capteur

Le problème principal est le calcul des paramètres  $b_{ij}$  de la transformation  $(i,j) \rightarrow (X,Y)$  (voir paragraphe 1.1). Sa résolution est faite en deux étapes. La première (traitement d'image) consiste à repérer les quatre points anguleux sur un objet étalon. La deuxième (calcul numérique) consiste à résoudre un système de 8 équations à 8 inconnues, où les 8 inconnues sont les paramètres  $b_{ij}$ . Ce calcul est fait simplement avec l'algorithme de Gauss.

Le repérage des quatre points est effectué par découpage récursif [Duda-73] de la ligne projetée sur l'objet étalon et fournie par le processeur GTR-3D sous la forme d'un vecteur  $CONTOUR(j)$ .

Le découpage récursif (voir figure 2.7) est effectué de la façon suivante. On dispose d'une courbe  $C(t):(X(t),Y(t)), t \in [1,n]$  où les points  $C(k)$  et  $C(k+1)$  sont deux points consécutifs (pour  $k \in [1,n-1]$ ). Si tous les points compris entre les points  $C(2)$  et  $C(n-1)$  ont une distance à la droite  $D$  passant par  $C(1)$  et  $C(n)$  inférieure à un seuil donné  $d$ , alors le découpage s'arrête. Sinon on prend parmi ces points le point d'indice  $c$ , dit point de segmentation, qui maximise la distance à la droite  $D$  et on appelle récursivement l'algorithme pour les portions  $[1,c]$  et  $[c,n]$  de la courbe.

Après le découpage récursif, on suppose que les quatre points anguleux recherchés font partie des points de segmentation. Mais il peut se faire qu'on obtienne d'autres points. Un algorithme basé sur la mesure des angles entre segments successifs élimine les points non significatifs (voir figure 2.8). On obtient alors les quatre points  $M_1, M_2, M_3$  et  $M_4$  recherchés. L'ensemble de ce traitement s'avère très fiable si l'on choisit un "bon" objet étalon présentant des points anguleux bien distincts (distance suffisante entre deux points consécutifs, angle entre deux segments successifs assez marqué).

Pour améliorer la précision, on calcule alors les meilleures droites au sens des moindres carrés qui passent par les points de  $CONTOUR(j)$  situés entre  $A$  et  $M_1$ , entre  $M_1$  et  $M_2$ , etc... et on calcule ensuite les intersections de ces droites entre elles, ce qui donne les points  $M_1', M_2', M_3', M_4'$  (voir figure 2.9). Ces différentes étapes sont visualisées sur un écran, voir photo 2.10.

Diverses primitives d'affichage graphiques permettent de visualiser les différents pas du traitement (voir photos 2.4 et 2.5).

### Remarque 1 :

Il est possible que dans le profil  $CONTOUR(j)$  se trouvent quelques points "aberrants". Ces points sont caractérisés par le fait qu'ils sont sans continuité avec les points qui précèdent et les points qui suivent. Pour les éliminer, on procède à un suivi de ligne basé sur le principe qu'un point n'est acceptable que s'il est proche d'un point acceptable soit à gauche, soit à droite, en utilisant la pente. Le processus est initialisé à partir de la recherche d'une suite d'au moins  $n$  pixels (15 dans notre application) consécutifs

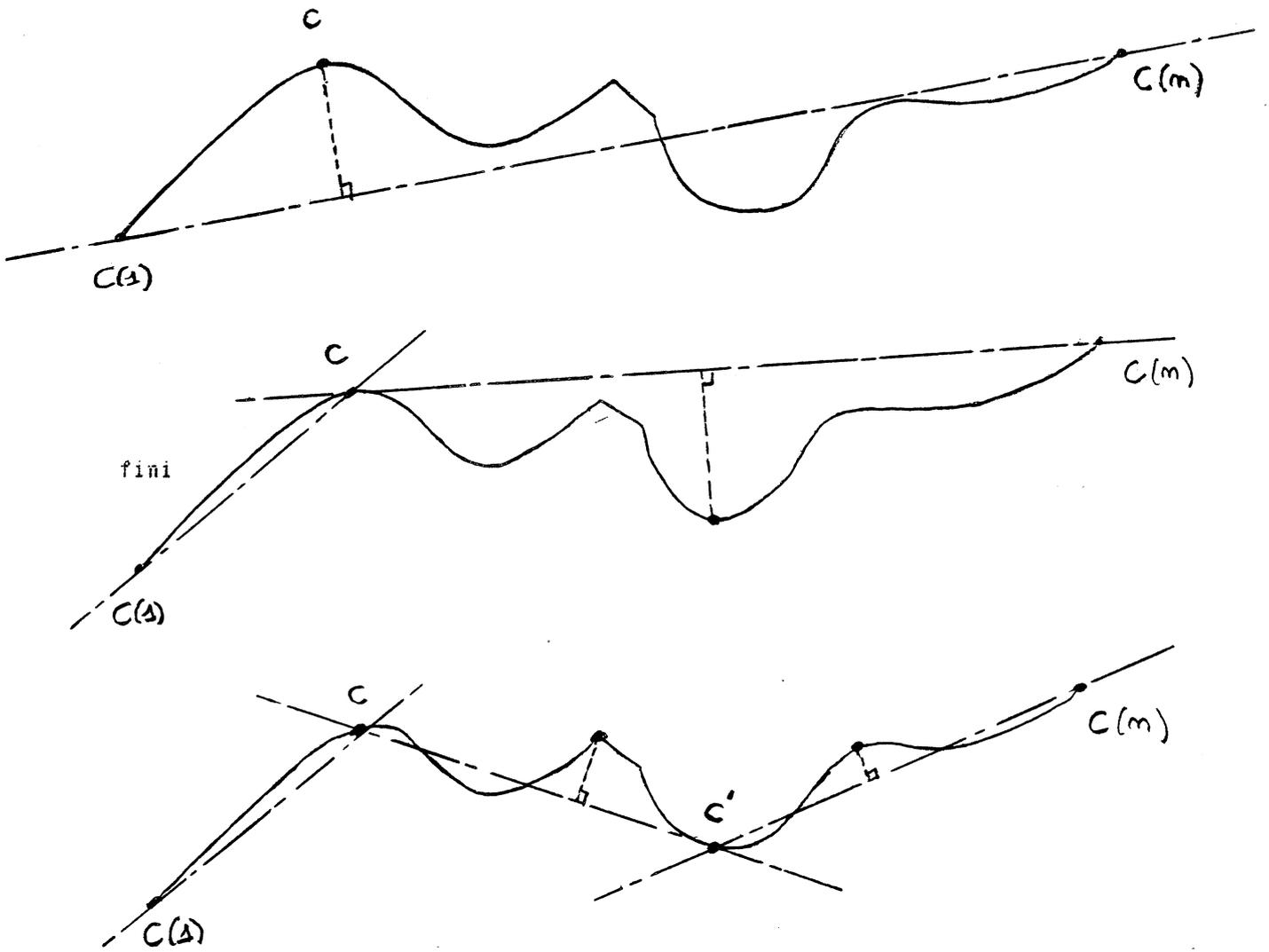


figure 2.7

Segmentation d'une courbe par découpage récursif.

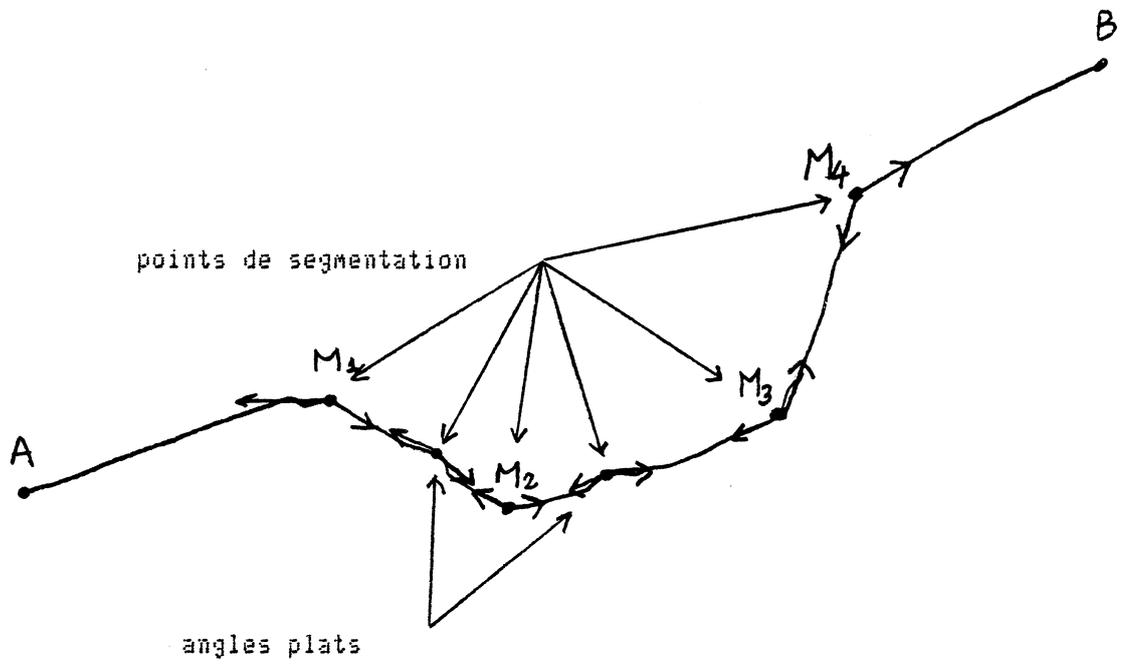


figure 2.8

Découpage de la ligne projetée sur un objet étalon.

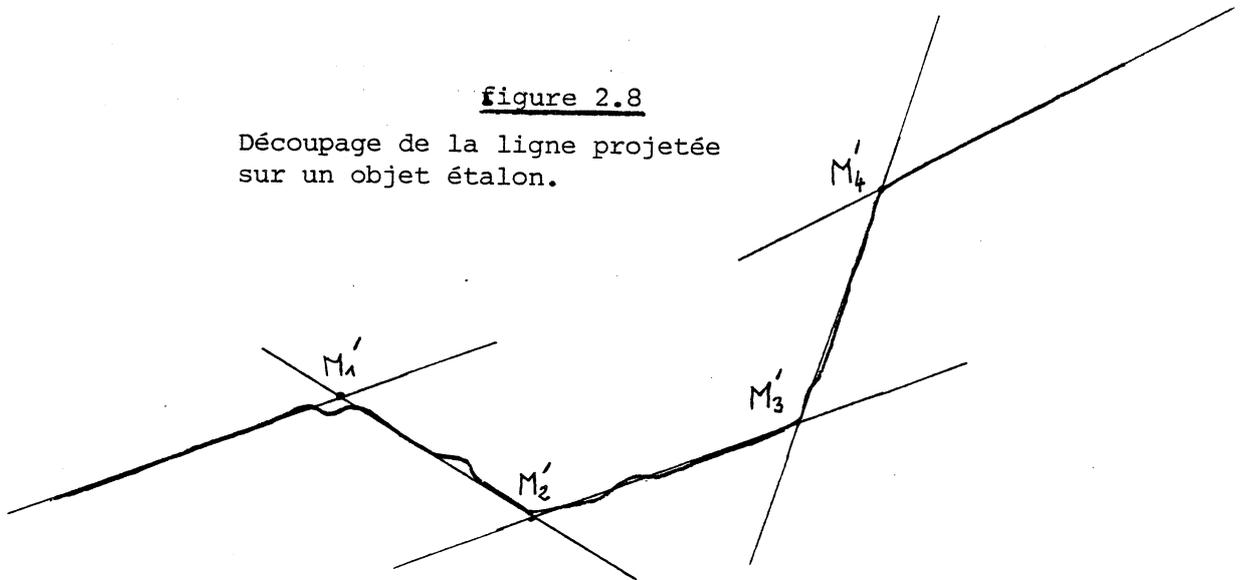


figure 2.9

Repérage des quatre points significatifs.

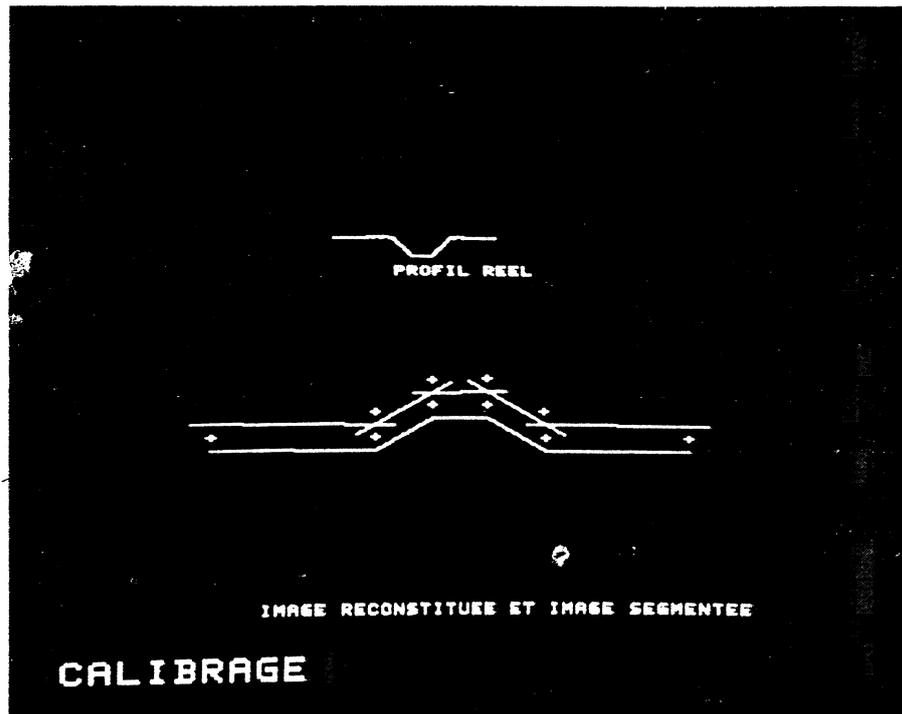


photo 2.10

Etapes du calibrage caméra - plan mumineux. Les points significatifs détectés sont marqués d'un "+" légèrement surélevés.

[Mézin-82].

Remarque 2 :

Suivant les applications il peut être intéressant d'utiliser divers objets étalons. On peut alors chercher à ce qu'un objet étalon occupe le plus de place possible dans l'image vidéo. Cette exigence peut se traduire de la manière suivante: il faut que chacun des quatre points significatifs se trouve à un sommet de l'image. Les objets étalons proposés à la figure 2.2 satisfont à cette exigence. Le traitement permettant de repérer les quatre points n'est pas le même que celui vu précédemment. Il est en effet nécessaire de repérer sur CONTOUR(j) des zones dites de "définition" maximales (au sens de l'inclusion) et des zones dites de "continuité" maximales (au sens de l'inclusion). Une zone [a,b] sera dite "définie" si tous les points situés entre a et b sont définis. Une zone [a,b] sera dite "continue" si elle est définie et si pour tout point k de [a,b] on a  $|\text{CONTOUR}(k) - \text{CONTOUR}(k+1)| < S'$ , ou  $S'$  est une valeur de seuil.

2.2 Construction d'image 3D

On dispose de la courbe "écran" donnée par  $(\text{CONTOUR}(j), j), j=1, 256$ . La courbe "espace" s'obtient en utilisant la fonction qui met en relation les points "écran" avec les points "plan lumineux", dont les paramètres ont été calculés au cours du calibrage. La courbe "espace" est donc la courbe:

$$\begin{cases} X(j) = \frac{b_{11} \text{CONTOUR}(j) + b_{12} j + b_{14}}{b_{31} \text{CONTOUR}(j) + b_{32} j + 1} \\ Y(j) = \frac{b_{21} \text{CONTOUR}(j) + b_{22} j + b_{24}}{b_{31} \text{CONTOUR}(j) + b_{32} j + 1} \end{cases} \quad j=1, 256$$

La troisième coordonnée,  $Z=tz$ , est liée à la position du plateau translatable au moment de la prise de vue, qui est connue par le système à chaque instant.

Comme nous l'avons vu dans la remarque 1 du paragraphe 1.2, les points  $(X(j), Y(j)), j=1, 256$  de la courbe "espace" ne sont pas régulièrement espacés, c'est-à-dire qu'on n'a pas  $X(i+1) - X(i)$  constant. Nous "reformatons" la représentation de cette courbe en construisant un vecteur  $f_1(X_k)$ , où  $X_k = X_0 + (k-1)\Delta X$ ,  $k=1$  à  $n_p$ , et  $\Delta X$  et  $n_p$  sont deux paramètres de la construction d'image (voir paragraphe 1.2). Cela se fait en deux pas de traitement:

- Le premier pas détermine dans quel intervalle  $[X(j), X(j+1)[$  se trouve chaque point  $X_0 + (k-1)\Delta X$ . Quand  $j$  n'existe pas (exemple pour le point  $X_0$ ), on convient de la valeur 0.
- Le second pas calcule  $f_1(X_k)$  par interpolation linéaire entre les valeurs  $Y(j)$  et  $Y(j+1)$ .

Chaque vecteur  $f_l(X_k)$  ainsi construit constitue une ligne de l'image  $Y=f(X,Z)$ .

### 3. QUELQUES ELEMENTS DE PERFORMANCE

#### 3.1 Vitesse

L'opération de base lors de la formation d'une image 3D est l'obtention d'un profil 3D. Elle peut être décomposée en:

- l'acquisition du profil (CONTOUR[1::256]),
- le redressement et l'échantillonnage du profil (Y(X)).

La première phase est effectuée par le processeur câblé GTR-3D et se fait en 1/50 ième de seconde. La seconde phase, logicielle, dure 1/2 seconde avec le calculateur LSI 11/23 utilisé.

Cette limitation à environ 2 images par seconde est due à ce que dans la configuration utilisée, le LSI 11/23 ne possède pas de processeur de calcul en virgule flottante capable d'effectuer des calculs sur des réels d'une manière câblée. Ces calculs sont donc effectués par micro-programme. Le nombre d'opérations nécessaires est de 256 fois 2 divisions, 6 multiplications et 6 additions. Nous avons évalué qu'avec un processeur câblé, on pourrait travailler à un rythme de 10 profils 3D à la seconde.

#### 3.2 Précision

Considérons la figure 2.11. On suppose qu'on a  $\vec{i}=\vec{X}$ , et que le point F est dans le plan  $X=0$ . L'application des équations (annexe D) permet d'écrire, si  $(i,j)$  sont les coordonnées d'un pixel et  $(X,Y)$  les coordonnées du point correspondant dans le plan  $Z=0$ :

$$\begin{cases} i = CF \frac{X}{-\cos\theta Y - OF} \\ j = CF \frac{-\sin\theta Y}{-\cos\theta Y - OF} \end{cases}$$

Les effets de perspective disparaissent si le point F est très

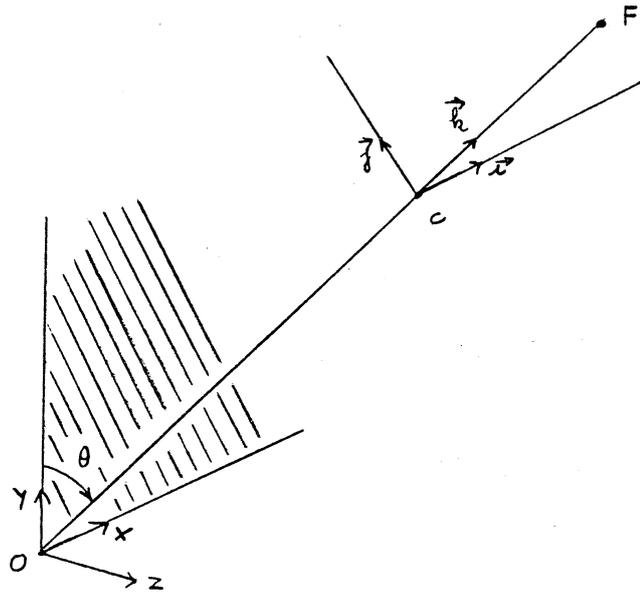


figure 2.11

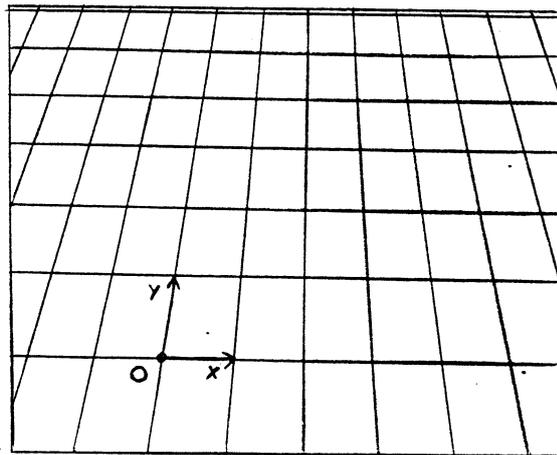


figure 2.12

éloigné, ce qui se traduit par  $OF \gg Y$ . On obtient alors:

$$\left\{ \begin{array}{l} i = - \frac{CF}{OF} X \\ j = - \frac{CF}{OF} \sin\theta Y \end{array} \right. \text{ ce qui donne } \left\{ \begin{array}{l} X = - \frac{OF}{CF} i \\ Y = - \frac{OF}{CF} \frac{j}{\sin\theta} \end{array} \right.$$

Si le champ de vision suivant l'axe X est de n cm, ce qui correspond à 256 pixels, on a alors:

$$\left\{ \begin{array}{l} \Delta X = \frac{n}{256} \Delta i \\ \Delta Y = \frac{n}{256} \Delta j / \sin\theta \end{array} \right.$$

Ce résultat confirme le fait que plus l'angle d'incidence ( $\theta$ ) de la caméra est proche de  $\pi/2$  et plus la précision est bonne ( $\Delta Y$  petit). Quand le champ de vision est de 4cm, et que l'angle  $\theta$  est de 45 degrés, on trouve  $4 \text{ cm}/256 = 0.16 \text{ mm}$  en X et  $4 \text{ cm}/(256 \cdot \sin 45) = 0.22 \text{ mm}$  en Y.

Soit un plan, sur lequel on trace deux réseaux de droites équidistantes, l'un de direction X et l'autre de direction Y. Toutes les "cases" de ce réseau sont équivalentes en surface. Considérons maintenant l'image qu'en donne une caméra (voir figure 2.12). L'image par la transformation projective de cases de même surface, donne des cases de surfaces différentes. Si donc on utilise une caméra pour inspecter ce plan, la précision de la mesure dépend de l'emplacement du point observé (plus on se rapproche des points de fuite et plus la précision est mauvaise). Il en est de même pour notre capteur, lorsque les effets de perspective ne sont pas négligeables. Dans ce cas, la précision est une fonction de i et de j.



CHAPITRE 3

QUELQUES APPLICATIONS DU CAPTEUR

1. Détection de défauts volumiques
2. Suivi de joints de soudure
3. Insertion de composants non standard



Dans ce chapitre nous présentons brièvement trois applications du capteur. Elles mettent en jeu des logiciels d'analyse d'images peu volumineux, car les données manipulées restent toutes conceptuellement proches des données issues du capteur. Cela ne signifie cependant pas que ces logiciels soient exempts de difficultés.

La première application est la détection de défauts volumiques, en creux ou en relief. Il s'agit de comparer l'image 3D d'une surface avec une image 3D de référence obtenue à l'aide d'un objet sans défaut; une application industrielle possible est l'inspection de moules de fonderie après démoulage. Ce travail a fait l'objet d'un projet d'étudiants de l'ENSIMAG [Drouin-83].

La deuxième application concerne le suivi de joints de soudure. Il s'agit de déterminer certains paramètres géométriques d'un profil d'un joint de soudure en vue de guider un robot. Ce travail a été réalisé dans le cadre d'un contrat avec la société COMEX.

La troisième application est l'insertion de composants électroniques non standard sur une carte de circuit imprimé. Le capteur 3D est utilisé pour localiser les "pattes" du composant et en déterminer leurs formes, afin de guider les mouvements d'un robot de montage. Cette étude fait actuellement l'objet d'un projet d'étudiants de l'ENSIMAG.

## 1. DETECTION DE DEFAUTS VOLUMIQUES

Le but des traitements est de contrôler la conformité géométrique d'une surface. Ils consistent en une comparaison de l'image 3D de la surface,  $Y=f_1(X,Z)$ , et d'une image 3D modèle,  $Y=f_0(X,Z)$ . La comparaison des deux images se fait après une procédure de recalage 2D en translation et en orientation. Cette procédure peut par exemple s'effectuer à l'aide de la détection de motifs dans chaque image (par exemple des trous) plus un algorithme de correspondance et de calcul de transformation. Ce recalage fournit une transformation T:

$$\begin{cases} X' = \cos\theta X + \sin\theta Z + X_0 \\ Z' = -\sin\theta X + \cos\theta Z + Z_0 \end{cases}$$

qui permet de transformer l'image  $f_1(X,Z)$  en une image  $f_2(X,Z)$ , recalée par rapport à  $f_0$ , avec  $f_2(X,Z)=f_1(T(X,Z))$ .

On calcule ensuite l'image différence en valeur absolue de  $f_0$  et  $f_2$ :  $D(X,Z)=|f_0(X,Z)-f_2(X,Z)|$ . Un seuil S permet de séparer les points "non conformes" à l'image de référence  $f_1$  des autres, ce qui nous fournit une image  $D_1(X,Z)$  telle que:  $D_1(X,Z)=D(X,Z)$  si  $D(X,Z)>S$ , 0 sinon.

Ce seul traitement ne suffit pas, en particulier si l'image de référence possède des discontinuités. En effet, un recalage même très précis fait apparaître ces zones de discontinuité dans l'image différence (voir figure 3.1). C'est pourquoi on applique à l'image différence après ce seuillage, un algorithme d'analyse de connexité

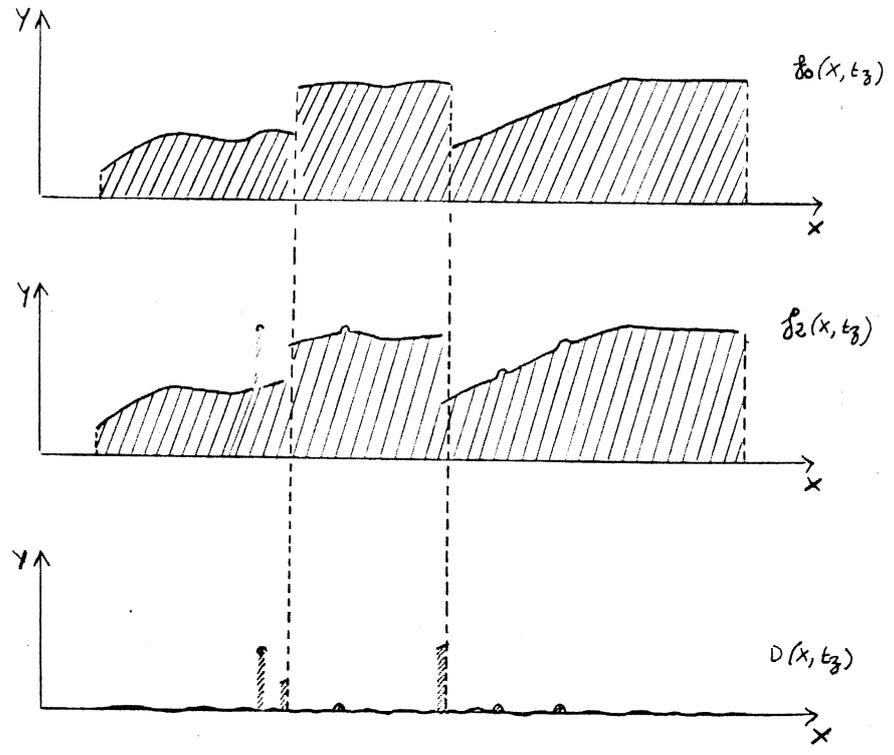


figure 3.1

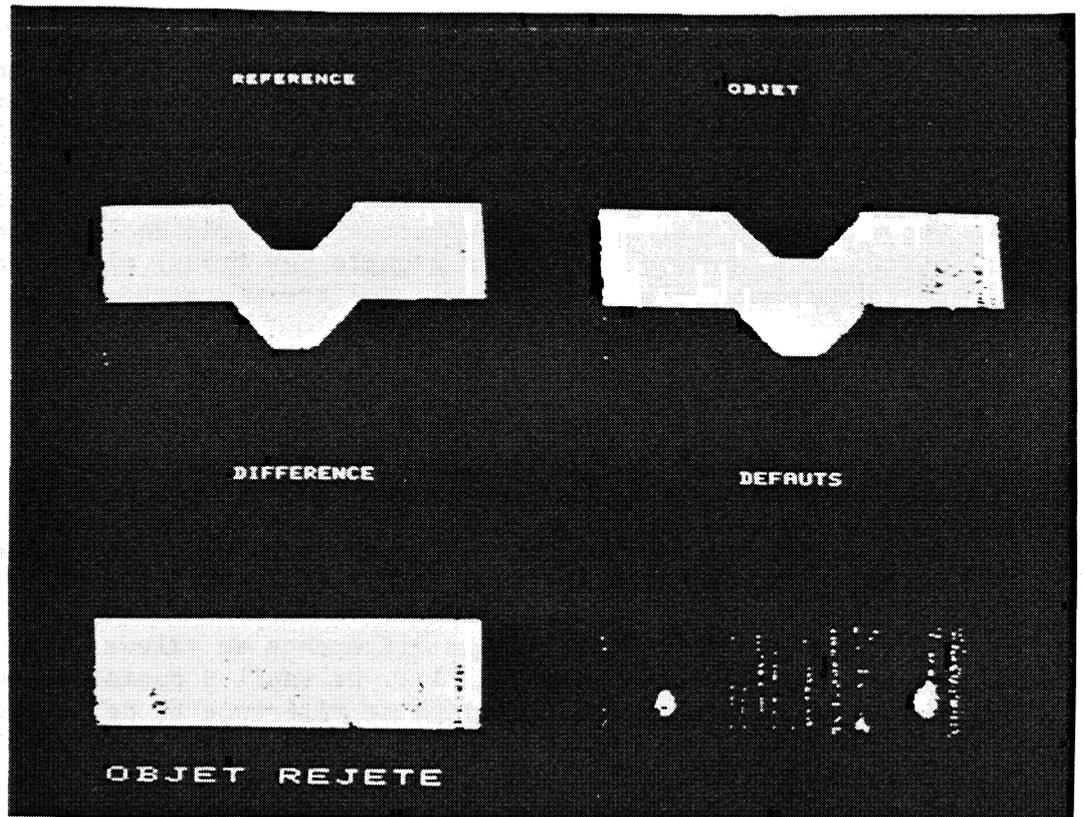


photo 3.2

[Veillon-77], qui regroupe en régions les points non conformes. A chaque région R est associé un volume de défaut égal à la sommation sur R des  $Dl(X,Z) \Delta X \Delta Z$ . Suivant la valeur du volume de défaut obtenu pour chaque région, et suivant la valeur totale du volume de défaut pour l'ensemble des régions, l'objet inspecté est rejeté ou accepté (voir photo 3.2).

## 2. SUIVI DE JOINTS DE SOUDURE

La photo 3.3 montre un exemple de joint de soudure. La soudure est effectuée en déposant des cordons successifs de métal chauffé par un arc à souder. Pour automatiser cette opération à l'aide d'un robot, il faut connaître certains paramètres géométriques du joint au fur et à mesure de la progression, de façon à ce que le robot positionne et oriente correctement le système de soudure. Pour le joint de la photo 3.3, les paramètres à déterminer sont typiquement la position des quatre points A,B,C,D montrés à la figure 3.4, le niveau moyen des cordons entre C et D, etc... Cette mesure doit être faite tous les 0,5 mm environ.

Le besoin d'un capteur pour suivre un joint de soudure peut avoir plusieurs causes, notamment:

- La position des pièces à souder n'est pas suffisamment bien connue. C'est le cas de la société COMEX qui s'intéresse à la soudure sur des pipe-lines sous-marins. C'est aussi le cas en atelier lorsque les pièces sont bridées avec peu de précision.
- L'échauffement dû à la soudure provoque une déformation des pièces à souder qu'il n'est pas possible de prévoir quantitativement; c'est le cas en particulier pour la soudure des tôles minces.

La détermination des paramètres géométriques sur les coupes successives d'un joint de soudure peut être faite en utilisant un capteur, soit lors d'une "passe" précédant la soudure proprement dite, soit en déplaçant le capteur avec le système de soudure. S'il peut y avoir déplacement relatif des objets à souder au cours de la soudure, la seconde solution est nécessaire.

Plusieurs types de capteurs peuvent être envisagés, capteur capacitifs, palpeurs mécaniques,... Toutefois, le capteur 3D décrit dans le chapitre 1 et 2 présente des avantages potentiels de vitesse et d'endurance.

L'expérimentation de notre capteur en présence de l'arc à souder a montré que les informations transmises sont "convenables", c'est-à-dire assez peu bruitées, en plaçant un cache entre l'arc et le capteur, descendant jusqu'à 1 à 2 cm du joint (voir figure 3.5). Le capteur est alors situé à environ 5 cm devant le système de soudure. Outre sa fonction optique, le cache sert aussi de cage de Faraday pour isoler électriquement le capteur, ainsi que de protection thermique et contre les éclats de soudure.

Malgré ce cache, un algorithme relativement sophistiqué de suivi de ligne dans une image a toutefois dû être réalisé pour éliminer les bruits restants. Cet algorithme combine des méthodes générales de suivi de ligne avec des connaissances a priori de la

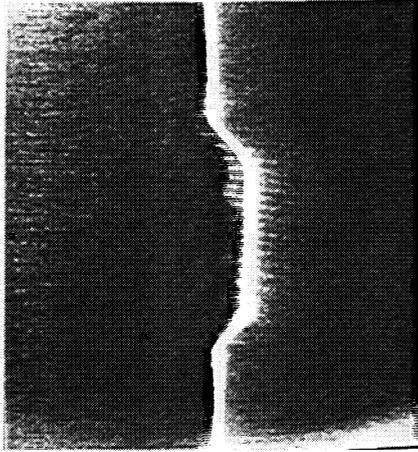


photo 3.3

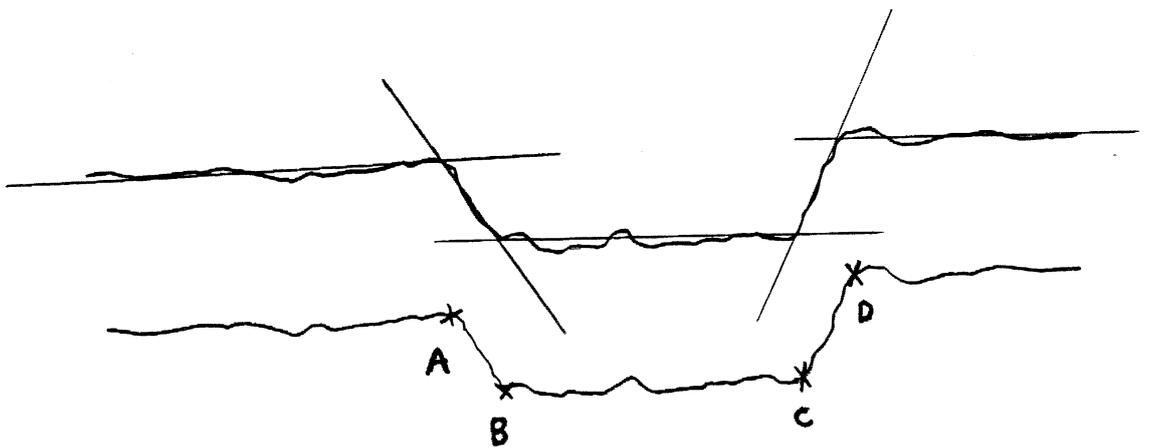


figure 3.4

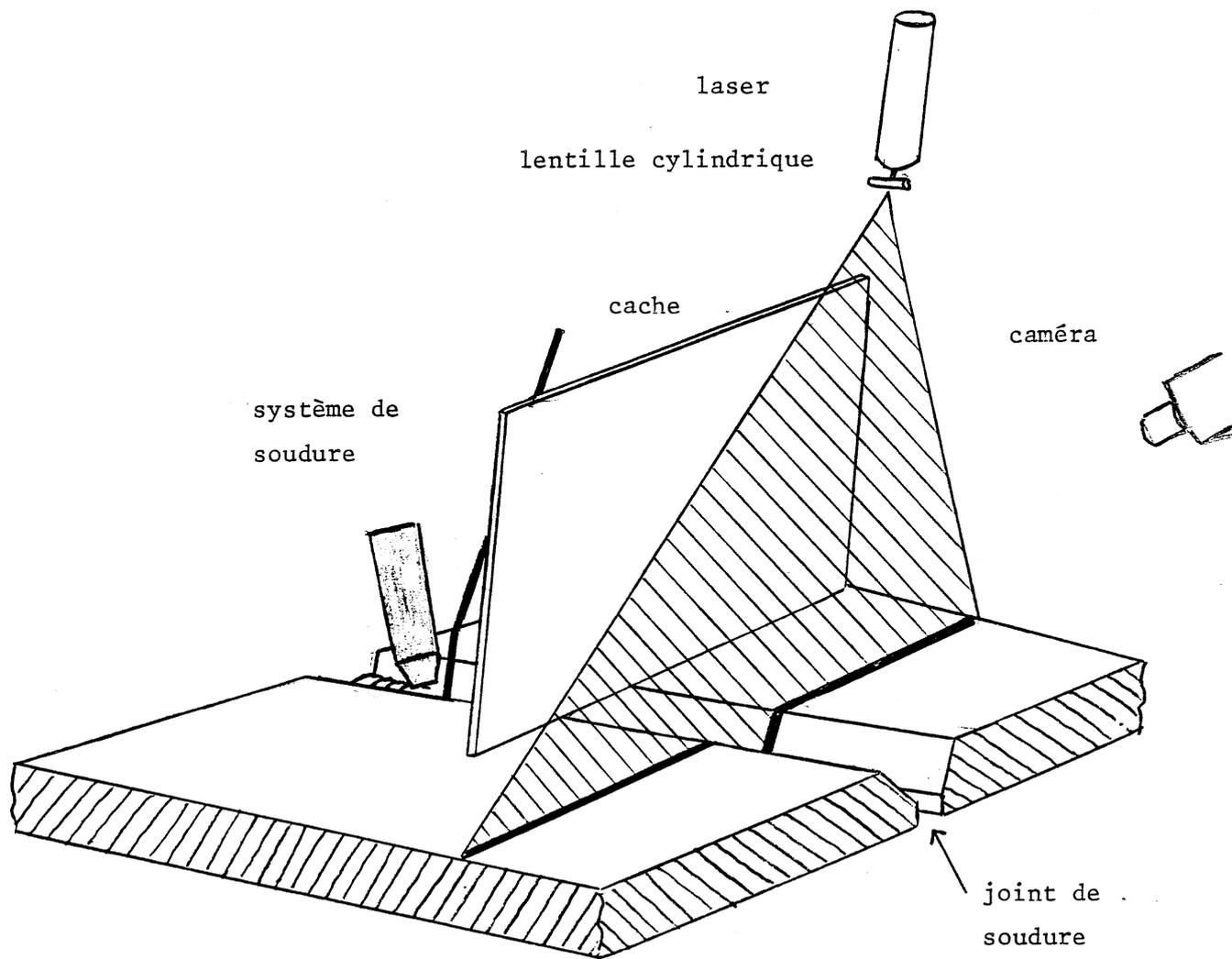


figure 3.5

géométrie type du joint et une comparaison des résultats sur le profil courant avec les résultats obtenus pour le profil précédent.

Muni de cet algorithme, le capteur permet d'obtenir un profil propre avec une bonne fiabilité en présence de l'arc. Les points A,B,C et D sont alors déterminés dans la courbe "image" par une variante de l'algorithme de découpage récursif présenté au paragraphe 2.1 du chapitre 2, qui utilise une connaissance a priori de l'emplacement des quatre points pour éliminer des points de segmentation parasites. Une fois les points repérés dans l'image, on calcule leurs coordonnées dans l'espace.

Parallèlement aux travaux présentés ci-dessus, plusieurs études du même type ont été effectuées dans d'autres laboratoires [Bamba-81] [Villers-82].

### 3. INSERTION DE COMPOSANTS NON STANDARD

Aujourd'hui seule l'insertion de composants électroniques standard est bien automatisée. Toutefois 20 à 30 % des composants sur une carte ne sont pas standard. Le but est ici d'utiliser les informations transmises par le capteur 3D pour guider les déplacements d'un bras manipulateur lors de l'insertion d'un composant non standard sur une carte. La photo 3.6 montre le robot tenant le composant, et le plan lumineux coupant les pattes du composant. On peut calculer les positions des points lumineux dans l'espace. Pour permettre les différents calculs, des procédures de calibrage lient au préalable les repères associés au système de vision, à la carte sur laquelle seront insérés les composants et au robot.

L'insertion d'une patte se fait alors de la manière suivante. Le robot positionne le composant pour que la patte coupe le plan lumineux. Le capteur fournit alors un point  $(X_1, Y_1, Z)$ , où  $Z$  est la hauteur du plan lumineux (sans importance dans notre cas). Le robot descend la patte suivant  $Z$ , vers le bas, d'une valeur  $t_z$  et le capteur nous fournit alors un second point  $(X_2, Y_2, Z)$ . Le vecteur  $V_1 = (X_1 - X_2, Y_1 - Y_2, t_z)$  donne l'orientation de la patte (si elle est verticale, on a  $(0, 0, t_z)$ ). On calcule une rotation qui permet de rendre verticale la patte, et le robot effectue le déplacement. Le capteur fournit un nouveau point  $(X_3, Y_3, Z)$ . La patte devant être insérée au point cible  $(X_c, Y_c, Z)$ , on calcule la translation de vecteur  $V_2 = (X_c - X_3, Y_c - Y_3, 0)$  qui permet d'amener la patte au dessus du trou, et le robot l'exécute. Un déplacement suivant l'axe  $Z$ , vers le bas, termine alors l'insertion.

Le travail actuellement en cours porte sur la prise en compte de plusieurs pattes simultanément. L'utilisation du capteur pour cette opération d'insertion permet de s'affranchir des problèmes de positionnement absolu des composants dans la pince du robot, et de contraintes sur l'orientation des pattes du composant. Il faut néanmoins que l'inclinaison des pattes ne soit pas trop grande (45 degrés au maximum) et qu'elles soient assez rectilignes (pour que la mesure de la pente ait un sens).

Une autre méthode d'insertion de composants développée par la compagnie IBM consiste à utiliser un capteur de force. Nous pensons

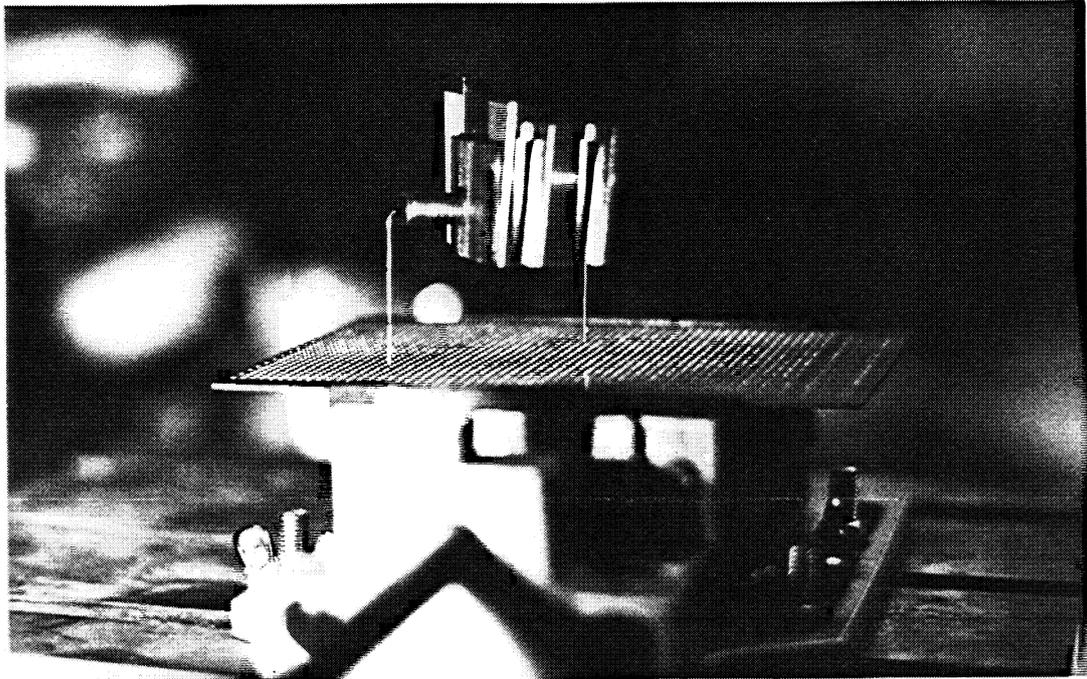


photo 3.6

que l'utilisation du capteur 3D peut permettre de rendre l'opération plus rapide; le capteur de force peut cependant être utilisé conjointement pour augmenter la fiabilité, le capteur 3D risquant de manquer de précision dans certains cas.

## CHAPITRE 4

## SEGMENTATION D'IMAGE 3D

1. Quelques méthodes de segmentation
  - 1.1 Transformée de Hough
    - 1.1.1 Principe
    - 1.1.2 Application aux images 3D
  - 1.2 Croissance de région
    - 1.2.1 Principe
    - 1.2.2 Application aux images 3D
  - 1.3 Découpage récursif
  - 1.4 Analyse de coupes
2. Notre méthode
  - 2.1 Cas des objets polyédriques
    - 2.1.1 Détection locale des plans
    - 2.1.2 Seuillage adaptatif
    - 2.1.3 Analyse de connexité
    - 2.1.4 Calcul des équations de plans
    - 2.1.5 Croissance de région
    - 2.1.6 Détermination des arêtes occultantes
    - 2.1.7 Autre exemple
  - 2.2 Cas des quadriques
3. Discussion



Le but de la segmentation d'une image est de réduire considérablement (typiquement par un facteur allant de 100 à 1000) la quantité d'informations contenues dans la matrice représentant l'image digitalisée, tout en dégagant les "entités" utiles aux traitements ultérieurs. Par exemple, la segmentation d'une image en intensité lumineuse peut consister à extraire les lignes de contraste, puis à découper ces lignes en segments de droites et arcs de cercle [Souvignier-83]; les traitements ultérieurs peuvent alors comparer les entités à des modèles bidimensionnels d'objets. En général, un algorithme de segmentation n'utilise pas de données a priori spécifiques d'une tâche, telles que des modèles d'objets. La nature des entités dégagées doit néanmoins être appropriée aux besoins des algorithmes qui les exploitent, et qui utilisent généralement des données spécifiques.

L'algorithme de segmentation que nous avons développé extrait d'une image  $Y=f(X,Z)$  des portions de plans et de quadriques, et fournit en sortie une description symbolique de l'image sous la forme d'une structure relationnelle de ces entités. Cette description est destinée à un programme de modélisation géométrique capable de construire le modèle d'un objet à partir de plusieurs images  $Y=f(X,Z)$  de cet objet sous différents angles de vue, en établissant des correspondances partielles entre les descriptions (voir chapitre 5). Elle peut aussi servir à identifier les objets présents dans la scène en établissant des correspondances avec des modèles a priori des objets.

L'application qui consiste à identifier des objets dans une scène a aussi donné lieu à d'autres segmentations. Par exemple, dans [Bolles-81] on extrait les arêtes d'une image 3D en utilisant un algorithme de "zero-crossing" développé pour des images d'intensité lumineuse [Hildreth-80], puis on découpe ces arêtes en segments droits et segments circulaires. Nous pensons que l'extraction de portions de surface fournit une information plus riche, puisqu'elle inclut l'information sur les arêtes.

Pour extraire des portions de plans et de quadriques d'une image 3D, plusieurs méthodes ont été développées: l'exposé de plusieurs d'entre elles fait l'objet du paragraphe 1 de ce chapitre. Relativement à ces méthodes, notre algorithme présente deux caractéristiques intéressantes:

- il comporte plusieurs pas de traitement successifs, dont la plupart, exécutables ligne par ligne, ont été imaginés pour faciliter une implantation en électronique câblée ou micro-programmée,
- il repose sur une interaction entre des calculs locaux (c'est-à-dire au niveau du pixel) pour détecter la présence des entités, et des traitements de confirmation globale, destinés à améliorer la fiabilité des résultats.

Le paragraphe 2 décrit en détail notre algorithme. Le paragraphe 3 le discute, relativement aux méthodes du paragraphe 1.

## 1. QUELQUES METHODES DE SEGMENTATION

Dans ce paragraphe nous présentons plusieurs méthodes utilisées pour extraire des portions de surface (plans et quadriques) d'une image 3D: transformée de Hough, croissance de région, découpage récursif, analyse de coupes.

### 1.1 Transformée de Hough

#### 1.1.1 Principe

La transformée de Hough permet de détecter la présence de formes  $F(X,P)=0$ , où  $P$  est l'ensemble des paramètres de cette forme, dans un image  $I(X)$ . Nous présentons son principe pour la détection des droites.

Soit  $G(i,j)$  une image binaire obtenue en seuillant le module du gradient calculé sur une image d'intensité lumineuse  $I(i,j)$ . Les pixels de valeur 1 dans  $G$  correspondent à des points de fort contraste dans  $I$ . On s'intéresse aux alignements de ces points.

L'équation d'un alignement est  $F(i,j,r,\theta)=i\cos\theta+j\sin\theta-r=0$ . La transformée de Hough fait passer de l'espace  $(i,j)$  à l'espace  $(r,\theta)$  des paramètres de la forme à détecter. On crée un tableau d'accumulation  $A[r_1::r_n;\theta_1::\theta_p]$  qui représente l'espace des paramètres discrétisé; ce tableau est initialisé à 0. Ensuite pour chaque pixel  $(i,j)$  de  $G$  ayant la valeur 1 on incrémente de 1 la valeur des cases  $(r_k,\theta_l)$  de  $A$  telles que  $F(i,j,r_k,\theta_l)<S$ , où  $S$  est un seuil donné. Une fois l'accumulation terminée, on interprète chaque case  $(r_k,\theta_l)$  de  $A$ , dont la valeur constitue un maximum local dans  $A$  supérieur à un seuil  $S'$ , comme étant la manifestation d'un alignement  $F(i,j,r_k,\theta_l)=0$  de points à fort contraste dans  $I(i,j)$ .

Cette méthode a d'abord été appliquée à la recherche de droites [Duda-72]. Elle a ensuite été étendue à la recherche de courbes plus complexes [Ballard-81].

Appliquée à la détection de courbes complexes, la transformée de Hough présente deux types de difficultés:

- Le choix de la discrétisation de l'espace des paramètres: un pas de discrétisation trop petit conduit à disperser l'accumulation résultant d'une même forme sur plusieurs cases de  $A$ ; un pas trop grand peut conduire à confondre plusieurs formes dans une même case.
- La taille du tableau d'accumulation: la dimension de  $A$  étant égale au nombre de paramètres inconnus des formes recherchées, la taille de  $A$  augmente très vite avec ce nombre.

Une solution possible à ces difficultés est de réaliser une discrétisation dynamique de l'espace des paramètres [O'Rourke-81].

### 1.1.2 Application aux images 3D

La transformée de Hough a été appliquée à la détection de formes planes et quadriques dans une image 3D à l'INRIA [Hébert-83] et au CRIN [Muller-84]. Un travail dans ce sens est également en cours au LETI.

[Hébert-83] caractérise une forme plane ou quadrique par ses normales sur la sphère Gaussienne [Ballard-82]. Par exemple, si l'on représente les normales à un plan par leurs extrémités sur la sphère unité, on obtient un point; les normales à une surface cylindrique donnent un cercle centré sur le centre de la sphère. La méthode pose toutefois deux difficultés. D'abord il faut choisir une discrétisation sur la sphère en cellules de taille constante facilement indexable. Ensuite, elle est difficilement applicable aux formes autres que les plans et les cylindres (taille du tableau d'accumulation et coût des calculs).

[Muller-84] détermine en chaque point de l'image 3D les paramètres de la meilleure quadrique passant par ce point et au moins 9 de ses voisins au sens des moindres carrés. Le problème étant linéaire pour les paramètres cherchés, la solution par les moindres carrés est simple. Les résultats servent à incrémenter un petit nombre de cases du tableau d'accumulation. La taille de ce tableau est gérée par une méthode de discrétisation dynamique qui subdivise les cases dont la valeur dépasse un certain seuil. L'évaluation des paramètres de quadrique en chaque point permet de maîtriser cette taille, et favorise la convergence vers les points d'accumulation.

Une implantation parallèle de la transformée de Hough est envisageable, l'algorithme étant le même pour tous les points de l'image. On notera toutefois que, telle quelle, la transformée de Hough ne donne pas l'emplacement des formes détectées dans l'image (celles-ci peuvent même être morcelées), et qu'un algorithme de "récupération" des formes est donc nécessaire en post-traitement.

## 1.2 Croissance de région

### 1.2.1 Principe

Le problème général de la segmentation en régions est défini de la manière suivante dans [Ballard-82]. Il s'agit de trouver une partition de l'image en régions connexes qui vérifie les deux relations suivantes:

$$\begin{cases} H(R_k) = \text{vrai, pour toute région } R_k \\ H(R_i \cup R_j) = \text{faux, pour tout couple } (R_i, R_j) \text{ de régions} \\ \text{adjacentes } (i \neq j). \end{cases}$$

où  $H$  est une fonction booléenne qu'on peut considérer comme un critère de fusion ("merging") (si  $H(R_i \cup R_j) = \text{vrai}$ ) ou comme un

critère de scission ("splitting") (si  $H(R_k)=\text{faux}$ ).

Une autre définition [Pavlidis-77] ramène le problème à trouver une partition de l'image en régions connexes, qui minimisent le critère suivant:

$$\sum_k E(R_k)$$

où  $E$  est une fonction qui évalue la "qualité" d'une région  $R_k$ .

Les algorithmes de croissance de région, dans le cas des images en niveaux de gris, partent des pixels comme régions élémentaires et procèdent par fusions successives.

Deux points importants conditionnent le fonctionnement de ce type d'algorithme:

- le premier est le critère de fusion de deux régions  $R_i$  et  $R_j$  ( $H$  dans la définition de Ballard)
- le second est l'ordre de fusion des couples de régions, c'est-à-dire le contrôle.

En particulier le choix du prochain couple à fusionner peut se faire sur une décision locale dans certains cas, ou globale dans d'autres. Les algorithmes fonctionnant avec un choix local sont sous-optimaux par rapport à ceux fonctionnant avec un choix global (dans la définition de Pavlidis), mais ils permettent des temps de calcul raisonnables.

Certains algorithmes combinent les processus de fusion et de scission. Ils nécessitent des structures représentant les régions et leurs frontières.

### 1.2.2 Application aux images 3D

#### a) Cas des objets polyédriques

La croissance de région a été utilisée pour segmenter des images 3D en régions planes par [Hébert-83].

Dans le cas 2D, les régions élémentaires de départ sont les pixels de l'image. En 3D, il s'agira soit des points  $(X,Y,Z)$  de la surface, soit des facettes obtenues par triangulation de l'objet 3D [Pauchon-83].

Hébert propose comme critère d'évaluation de la "qualité" de l'approximation d'une région  $R$  par un plan l'écart moyen  $E(R)$  des points au plan  $P$  calculé avec les moindres carrés sur cette région. Il serait possible d'ajouter des limitations sur les écarts maxima à ce plan.

En ce qui concerne la stratégie de choix des couples de régions  $(R_i, R_j)$  candidats à la fusion, il existe plusieurs possibilités. Dans tous les cas, le calcul est fait pas à pas, sans retour arrière.

La première possibilité consiste à fusionner les deux régions  $R_i$  et  $R_j$  qui minimisent l'erreur totale  $\sum E(R_k)$  après fusion. Son inconvénient majeur est qu'il faut parcourir l'ensemble de tous les couples de régions à chaque itération (c'est un contrôle global). La seconde et la troisième possibilité présupposent qu'on ne fasse pas la croissance en fusionnant des régions, mais en ajoutant progressivement des points aux régions. Pour chaque région  $R_i$  à un moment donné, on détermine la liste  $L_i$  des points adjacents à son contour. La seconde possibilité consiste à adjoindre à  $R_i$  le premier point  $P_k$  de  $L_i$  tel qu'on ait  $E(R_i \cup \{P_k\}) < T$ , où  $T$  est un seuil donné, c'est un contrôle local. La troisième possibilité consiste à adjoindre à  $R_i$  le point  $P_k$  de  $L_i$  qui minimise  $E(R_i \cup \{P_k\})$  et tel qu'on ait  $E(R_i \cup \{P_k\}) < T$ . Cette dernière possibilité est plus longue en temps de calcul que la seconde, car il faut parcourir la liste  $L_i$  dans son ensemble, bien qu'il soit possible d'optimiser les temps de calcul. En effet, en triant la liste  $L_i$  suivant le critère  $E$ , on peut supposer que l'ordre obtenu ne change pas sensiblement tant que la région n'a pas augmenté suffisamment de surface. Il est donc possible de ne faire ce tri que périodiquement.

#### b) Cas des quadriques

Shirai utilise la croissance de région pour segmenter des images 3D en portions de plans et quadriques.

Dans une première partie, il segmente l'image 3D en régions planes. Sa méthode est assez proche de celle utilisée par Hébert. L'initialisation d'un processus de croissance se fait par le choix d'un élément de surface satisfaisant un critère global. Ensuite un critère local permet d'ajouter de nouveaux éléments de surface à une région donnée. Quand tous les éléments de surface adjacents à la région ne répondent plus à ce critère, la croissance de cette région est terminée. On voit donc que le contrôle est local, sauf en ce qui concerne l'initialisation.

Shirai classe ensuite chaque région obtenue en région "plane", "courbe" ou "indéfinie", et cherche à fusionner les régions "courbes", ce qui constitue un nouveau processus de croissance, plus global. Le critère de fusion est basé sur l'angle que font les deux régions entre elles.

La troisième partie consiste à approcher chaque région par une quadrique.

### 1.3 Découpage récursif

Nous utilisons cette méthode [Duda-73] en deux dimensions pour effectuer le calibrage du capteur (voir chapitre 2, paragraphe 2.1). Il existe une généralisation qui transpose le problème de l'approximation polygonale d'une courbe 2D au problème de l'approximation polyédrique d'une surface 3D [Boissonat-81] [Faugeras-82].

Le problème, dans le cas 2D, est relativement simple, pourvu qu'il existe une relation explicite ou implicite qui lie les points consécutifs de la courbe: par exemple si la courbe est  $C(t):(X(t),Y(t))/t \in [1,n]$ , alors  $C(k)$  et  $C(k+1)$  sont deux points consécutifs. Dans ce cas, le point de segmentation  $c$  segmente la courbe en deux courbes  $C_1(t)=C(t)/t \in [1,c]$ , et  $C_2(t)=C(t)/t \in [c,n]$ , auxquelles on va appliquer le même algorithme de découpage. Par contre, si les points de la courbe  $C$  sont donnés au hasard, l'algorithme ne peut opérer, une relation de connexité entre les point étant nécessaire.

Dans le cas 3D, cette relation est aussi nécessaire, qu'elle soit donnée explicitement (chapitre 1, paragraphe 3) ou implicitement (image  $Y=f(X,Z)$ ). Lors d'un appel récursif de l'algorithme, on dispose d'une portion de surface  $S$  définie par un ensemble de points et leurs relations de connexité et de trois points  $P, Q$  et  $R$  situés sur le pourtour, non adjacents et non alignés (voir figure 4.1). Ces trois points définissent une équation de plan. On détermine le point  $C$  de  $S$  le plus éloigné du plan  $PQR$ . Si la distance de  $C$  au plan est inférieure à un seuil donné, le découpage s'arrête et la portion de surface  $S$  est approchée par le triangle  $PQR$ . Sinon, il faut scinder la surface  $S$  en trois portions de surface  $S_1, S_2$  et  $S_3$  correspondant à  $PCQ, QCR$  et  $RCP$  (voir figure 4.1).

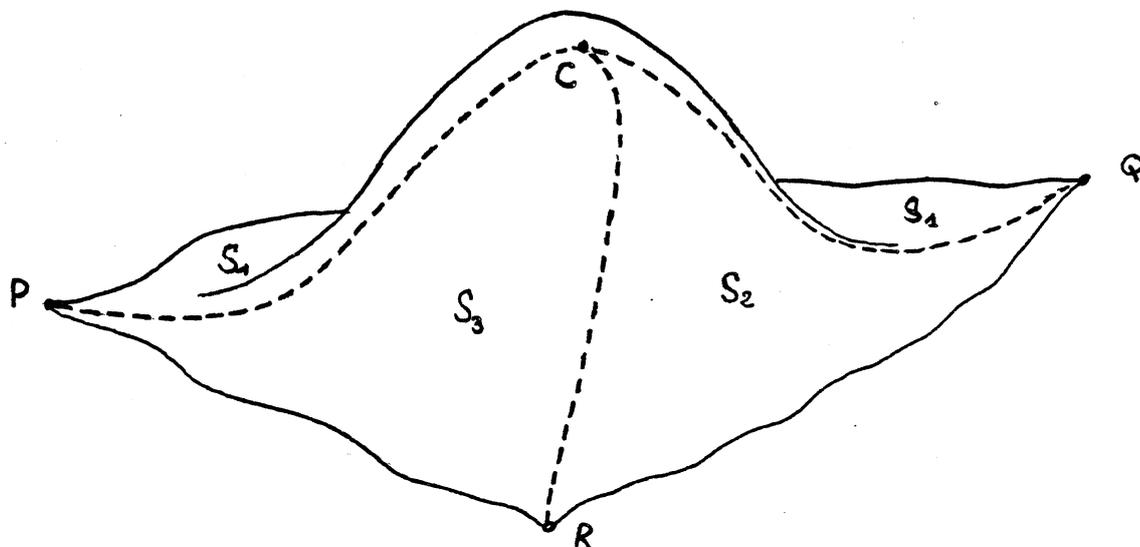


figure 4.1

Cette scission est effectuée en prenant le chemin (PC) (respectivement (QC) et (RC)), à l'intersection de S et du plan bissecteur des plans PCR et PCQ (respectivement: QCR et QCP, RCP et RCQ), à la discrétisation près. Les chemins (PC), (QC) et (RC) servent de frontières aux portions de surfaces  $S_1$ ,  $S_2$  et  $S_3$ . Ensuite, on appelle récursivement l'algorithme avec la surface  $S_1$  et les points P,C et Q; la surface  $S_2$  et les points Q,C et R; la surface  $S_3$  et les points R,C et P.

L'initialisation, dans le cas d'une surface fermée, consiste à prendre trois points P,Q,R sur la surface, non adjacents et non alignés. Le plan (PQR) permet de séparer la surface en deux surfaces  $S_1$  et  $S_2$ . On appelle l'algorithme de découpage avec la surface  $S_1$  et les points P,Q et R, puis avec la surface  $S_2$  et les points P,Q et R. Dans le cas d'une surface non fermée, on prend trois points P,Q et R sur son contour, non adjacents et non alignés. On appelle l'algorithme précédent avec la surface S et les points P, Q et R.

S'il existe, à un certain moment, une face parallèle au plan PQR, le point le plus éloigné sera l'ensemble des points de la face et le procédé risque de choisir un point de segmentation situé au milieu de cette face. C'est pourquoi on peut ajouter à cet algorithme une phase de fusion qui recombine les éléments séparés par erreur.

Cet algorithme fonctionne du global vers le local. La conservation de l'ordre de cette segmentation permet d'obtenir une structure hiérarchique sur les faces, mais qui est très dépendante du choix des trois points initiaux.

On notera la sensibilité de l'algorithme à la présence éventuelle de points "aberrants", qui peuvent perturber fortement la segmentation.

### 1.4 Analyse de coupes

Cette méthode est spécifique du type de notre capteur 3D. Elle consiste à analyser successivement les coupes CONTOUR[1::256] qu'il fournit.

On peut par exemple découper chaque coupe et retenir les segments droits. Une région plane dans l'image 3D se manifestera par des segments droits parallèles, équidistants et connexes dans plusieurs coupes successives.

Cette méthode a été développée et expérimentée dans le cadre d'un projet d'étudiant de l'ENSIMAG [Andrieux-83]. Elle a fourni des résultats satisfaisants sur des objets polyédriques. Toutefois la segmentation d'une coupe ne fournit pas toujours des segments de droite avec des extrémités fiables. Des post-traitements sont donc nécessaires.

On peut suivant le même principe détecter des portions de surfaces cylindriques. On travaille alors sur les coupes  $Y=f_1(X)$  dans lesquelles on recherche des portions d'ellipses. Mais les études faites [Andrieux-83] et [Bolles-81] ont montré que la position du centre d'une ellipse est assez mal définie par une portion de cette ellipse, lorsque ces points sont tous disposés du même côté.

La généralisation de la méthode à d'autres quadriques paraît encore plus problématique.

On notera que l'analyse de coupes par recherche de points de discontinuité et de points anguleux a aussi été utilisée pour extraire des arêtes dans une image 3D [Bolles-81] [Sugihara-77].

## 2. NOTRE METHODE

Nous présentons notre méthode en deux parties :

- Dans la première partie (paragraphe 2.1), nous la présentons dans un cas simplifié: celui où les objets sont polyédriques; les portions de surface qu'on doit extraire de l'image sont alors toutes des portions de plans délimitées par des arêtes droites.
- Dans la deuxième partie (paragraphe 2.2), nous généralisons les traitements décrits précédemment pour aborder le cas d'objets dont les enveloppes sont constituées de portions de plans et de portions de quadriques.

Cette division permet d'introduire les difficultés progressivement. Elle correspond aussi à une séparation entre ce que nous avons implanté et ce qui ne l'est pas encore; en effet seul le cas des objets polyédriques est actuellement traité par nos programmes.

L'implantation de la méthode restreinte au cas d'objets polyédriques a été faite en FORTRAN sur un ordinateur LSI 11/23.

## 2.1 Cas des objets polyédriques

La méthode que nous avons développée comporte six pas de traitement, à partir de l'image 3D originale (voir photo 4.2 (a)):

### 1 - Détection locale des plans (voir photo 4.2 (b))

L'image est transformée en une nouvelle matrice de même dimension, dont chaque point est une mesure locale de la "présence" d'un plan.

### 2 - Seuillage adaptatif (voir photo 4.3)

A l'aide d'un histogramme sur la matrice obtenue au pas précédent, on détermine un seuil qu'on utilise alors pour séparer les plans des arêtes.

### 3 - Analyse de connexité (voir photo 4.4)

On regroupe les points supposés appartenir à des plans en régions connexes. Chaque région constitue pour la suite des traitements l'"ébauche" d'une portion de plan.

### 4 - Calcul des équations de plans

On évalue pour chaque région connexe l'équation du meilleur plan au sens des moindres carrés.

### 5 - Croissance de région (voir photos 4.5 et 4.6)

On étend les régions connexes en utilisant les équations de plans et on construit les régions "arêtes" et "sommets".

### 6 - Détermination des arêtes occultantes (voir photo 4.7)

On traite le cas des arêtes occultantes des portions de plan (non traité par le pas 5).

#### 2.1.1 Détection locale des plans

Il s'agit d'effectuer une mesure locale de la présence d'un plan. Le résultat de cette opération est une nouvelle image  $P=g(X,Z)$  où chaque point  $(X,Z)$  contient une valeur  $P$  représentative de la présence d'une zone plane en ce point : plus la valeur de  $P$  est faible, plus la région centrée sur  $(X,Z)$  peut être assimilée à un plan; on aura donc des valeurs fortes de  $P$  pour les arêtes (voir photo 4.2).

Cette mesure se fait sur une fenêtre centrée sur le point pour lequel on fait le calcul, et dont on peut régler la taille (3 X 3, 5

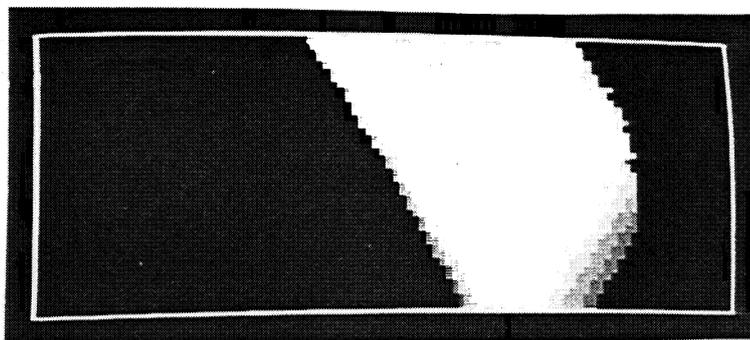


photo 4.2 (a)

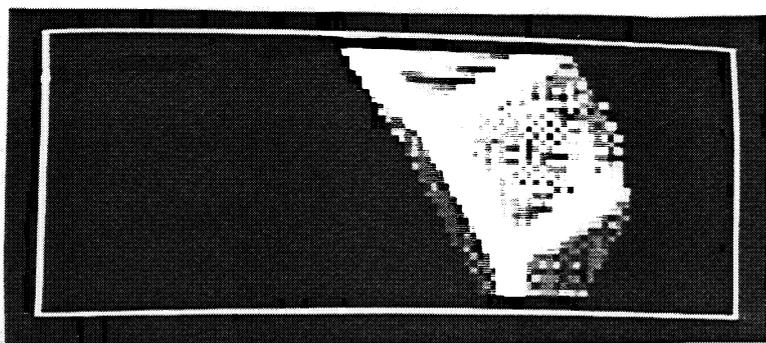


photo 4.2 (b)

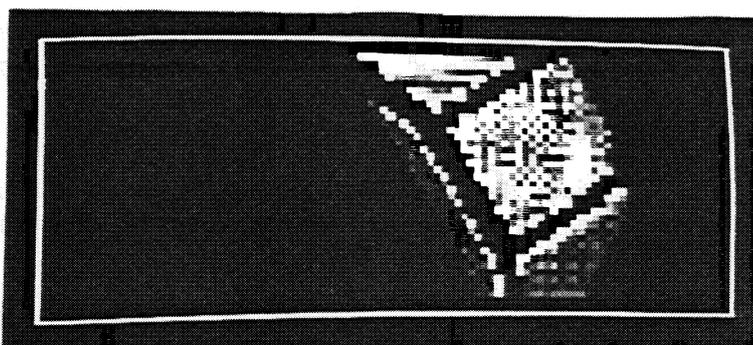


photo 4.3

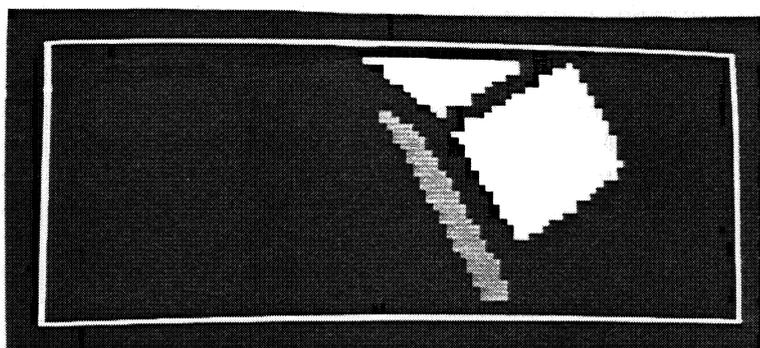


photo 4.4

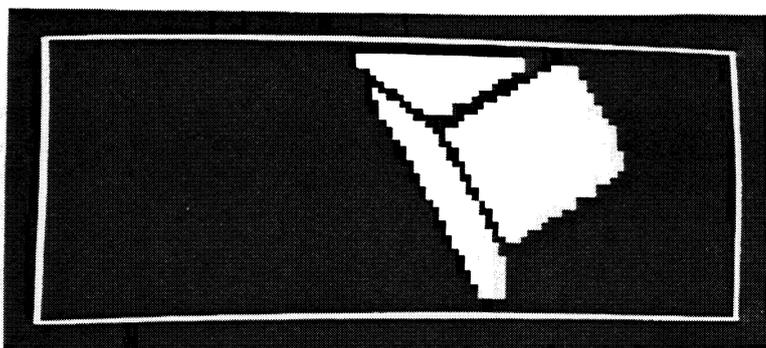


photo 4.5

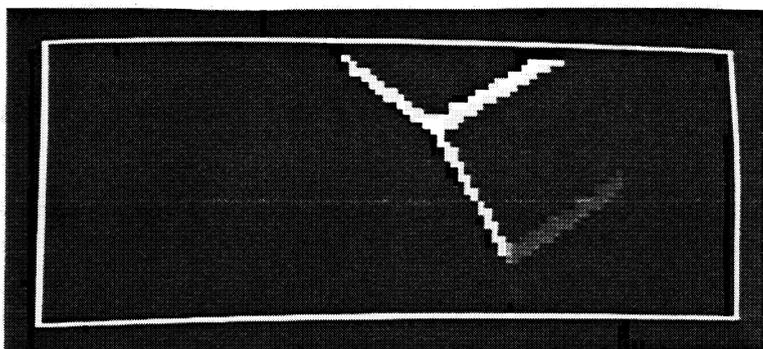


photo 4.6

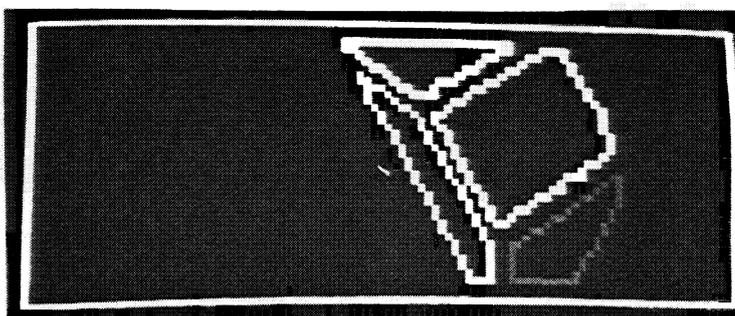


photo 4.7

X 5, etc ...). Il existe différentes méthodes pour l'obtenir. Nous en citerons deux :

- La première consiste à utiliser un procédé de convolution avec un opérateur gradient classique [Andrieux-83]. En effet, les régions planes sont celles où le vecteur gradient  $\nabla f$  est constant. Par conséquent le vecteur "double gradient"  $\nabla^2 f$  y est nul.

$$\nabla f : \begin{pmatrix} \frac{\partial f}{\partial X} \\ \frac{\partial f}{\partial Z} \end{pmatrix} \quad \nabla^2 f : \begin{pmatrix} \frac{\partial^2 f}{\partial X^2} \\ \frac{\partial^2 f}{\partial Z^2} \end{pmatrix}$$

La norme du vecteur "double gradient" est faible pour les plans et forte pour les arêtes. Elle constitue donc une mesure possible de la présence de zones planes. La photo 4.8 montre le résultat de son application. Nous n'avons pas choisi cette méthode pour deux raisons :

- Elle fait appel à des dérivées secondes, ce qui la rend sensible au bruit.
- Elle ne peut pas être généralisée à la détection d'autres formes que les plans, notamment les quadriques.

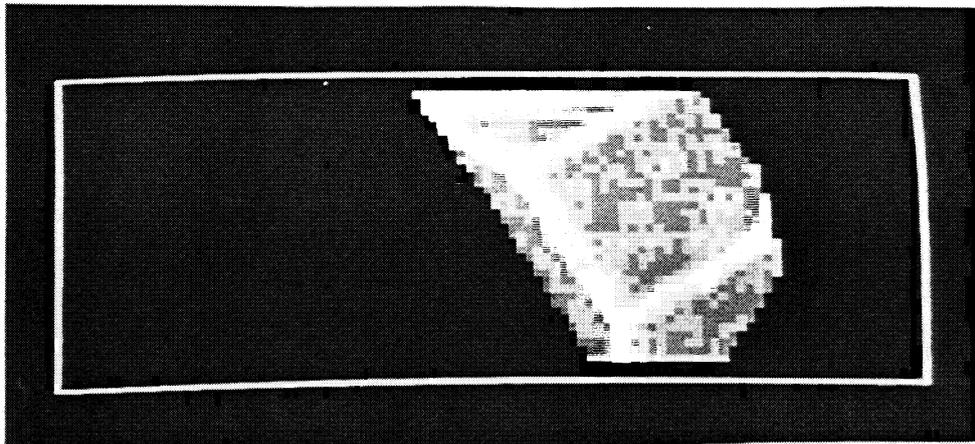


photo 4.8

- La seconde méthode, qui celle que nous avons choisie consiste à calculer en chaque point (X,Z) le meilleur plan au sens des moindres carrés.

Partant d'une image  $Y=f(X,Z)$ , on peut définir un plan par l'équation :

$$Y = a X + b Z + C$$

Les paramètres  $a$ ,  $b$  et  $c$  ne nous intéressent pas directement, car c'est l'écart quadratique résultant de l'approximation qui va être utilisé. Celui-ci est défini par :

$$M = \text{Min}_{a,b,c} \sum_{i=1}^n (Y_i - (a X_i + b Z_i + c))^2$$

où  $n$  est le nombre de points utilisés pour l'approximation (9 pour une fenêtre 3 X 3). La valeur :

$$P = \frac{M}{n}$$

représente l'écart moyen des points  $(X_i, Y_i, Z_i)$  au "meilleur" plan. Cet écart est d'autant plus faible que l'ensemble des points est proche d'un plan; on a donc ainsi une mesure de la présence de zones planes. Dans le cas où les points  $P_i$  forment un fenêtre 3X3, il est possible d'optimiser le calcul des coefficients  $a$ ,  $b$ ,  $c$  et de l'écart  $P$ . Les détails de ce calcul sont donnés dans l'annexe E.

Cette seconde méthode peut être généralisée aux quadriques, en substituant à l'équation de plan l'équation générale des quadriques.

#### Remarque sur l'implantation :

Notre capteur permet de choisir la résolution spatiale des images  $Y=f(X,Z)$  qu'on souhaite construire. De plus, on peut avoir une résolution suivant l'axe des  $X$  différente de la résolution suivant l'axe des  $Z$  (pixels non carrés). C'est pourquoi le logiciel implanté permet à l'utilisateur de définir la taille de la fenêtre du calcul de  $P$ , non pas en nombre de pixels, mais en dimension géométrique ( $N$  millimètres X  $N$  millimètres). L'utilisateur peut ainsi simultanément ignorer la résolution de son image et tenir compte des dimensions des objets pour choisir la valeur de  $N$ .

La fenêtre qui est effectivement utilisée pour le calcul de  $P$  est une approximation de celle définie par l'utilisateur de telle sorte qu'elle recouvre un nombre entier de pixels de l'image  $Y=f(X,Z)$ ; de plus chaque côté de cette fenêtre utilisée correspond à un nombre impair de pixels. Le calcul de  $P$  est alors effectué en utilisant 9 pixels de cette fenêtre. Notre logiciel se ramène ainsi toujours au cas d'un calcul sur une fenêtre 3 pixels X 3 pixels, pour lequel nous avons optimisé les calculs.

On pourrait encore améliorer les traitements en permettant de définir la taille de la fenêtre non pas dans le plan (X,Z), mais dans le plan tangent au point considéré. Les calculs seraient alors nettement plus longs.

### 2.1.2 Seuillage adaptatif

Le but de ce pas de traitement est de séparer les plans des arêtes. L'image  $P=g(X,Z)$  calculée au pas précédent contient l'information nécessaire, la valeur de  $P$  étant faible pour les plans et forte pour les arêtes. Le traitement effectué ici consiste à faire un seuillage de cette image.

L'histogramme sur les valeurs de  $P$  dans l'image  $g(X,Z)$  a typiquement l'allure montrée à la photo 4.9, avec deux maxima bien distincts correspondant aux plans et aux arêtes. Soit  $S$  la valeur de  $P$  correspondant au premier maximum. On considère alors que les points de l'image tels que  $P < S + d$  sont situés sur des zones planes. La valeur de  $d$  est une constante positive telle que  $S + d$  soit inférieur à la valeur de  $P$  correspondant au minimum de l'histogramme situé entre les deux maxima. Le résultat du seuillage est une image binaire (par convention les pixels considérés comme appartenant à une zone plane ont la valeur 1, les autres la valeur 0) telle que celle montrée à la photo 4.3.

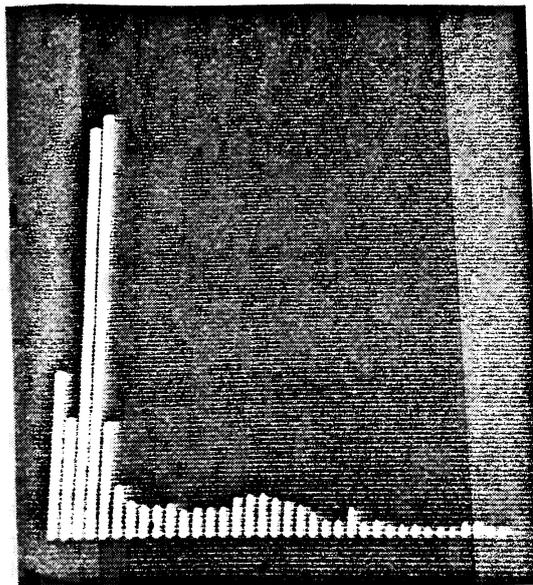


photo 4.9

La valeur de  $d$  doit être suffisamment faible si l'on ne veut pas que les différentes régions que l'on veut séparer au pas suivant par analyse de connexité, se trouvent connectées malencontreusement par des "ponts" (voir figure 4.10). Un choix trop faible risque d'éliminer des points appartenant à des zones planes, mais le pas 5 (croissance de région) de la méthode corrige ce mauvais classement.

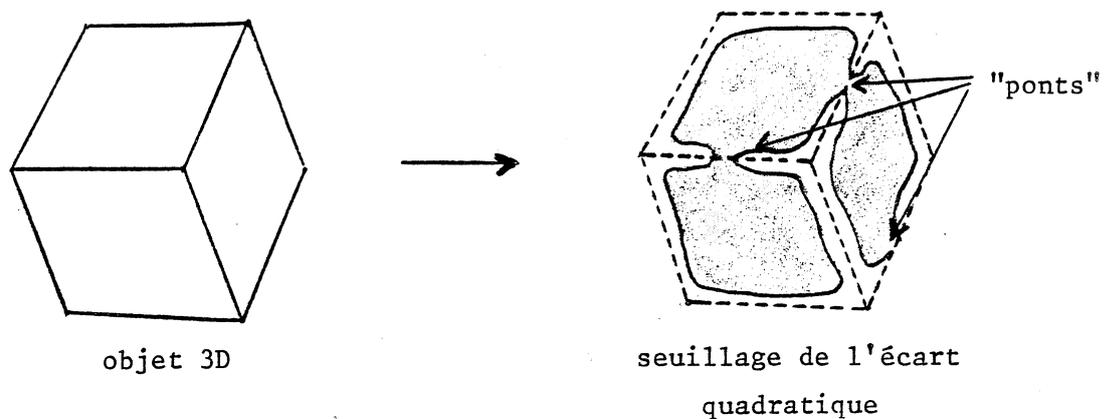


figure 4.10

Les expériences effectuées ont permis de vérifier que le choix du seuil  $S + d$  à partir de l'histogramme ne présente pas de réelle difficulté.

### 2.1.3 Analyse de connexité

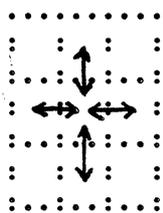
Jusqu'à présent, aucune donnée globale n'a été utilisée ou engendrée, si ce n'est l'histogramme nécessaire au choix du seuil de binarisation de  $g(X,Z)$ . Le second pas a mis en évidence les points que l'on suppose correspondre à des portions de plans. Il est maintenant nécessaire, pour accéder à des données plus globales, de connecter ces différents points en régions connexes maximales.

On dit que deux pixels A et B d'une image sont connexes si et seulement si il existe une suite  $x_0, x_1, \dots, x_n$  de pixels adjacents tels que  $x_0 = A$ ,  $x_n = B$  et  $n \geq 1$ . Lorsqu'on considère des pixels rectangulaires ce qui est notre cas, l'adjacence peut être définie de deux façons différentes :

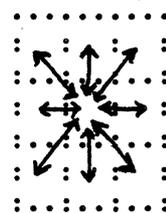
- deux pixels sont adjacents si et seulement si ils partagent un même côté (cf figure 4.11) : on parle alors de 4-connexité.
- deux pixels sont adjacents si et seulement si ils partagent un même côté ou un même sommet (cf figure 4.12) : on parle alors de 8-connexité.

Dans une image binaire, une région connexe constitue un ensemble de pixels connexes et de même valeur. La région est maximale si elle ne peut pas être entièrement incluse dans une autre région connexe. On remarque que si on construit les régions de pixels à 1 avec la 4-connexité, les régions de pixels à 0 sont

implicitement construites avec la 8-connexité, et réciproquement.



4-connexité

figure 4.11

8-connexité

figure 4.12

Nous formons les régions connexes maximales correspondant aux pixels à 1 dans l'image  $g(X,Z)$  seuillée en utilisant la 4-connexité. Nous procédons en utilisant un algorithme connu de connexité en un passage [Veillon-77] qui simultanément construit les régions et en calcule quelques propriétés (surface, centre de gravité). A l'issue de cet algorithme, on dispose du nombre de régions connexes, de leur définition sous forme d'une liste de pixels les constituant, de leurs importances respectives (surfaces) et de leurs centres de gravité.

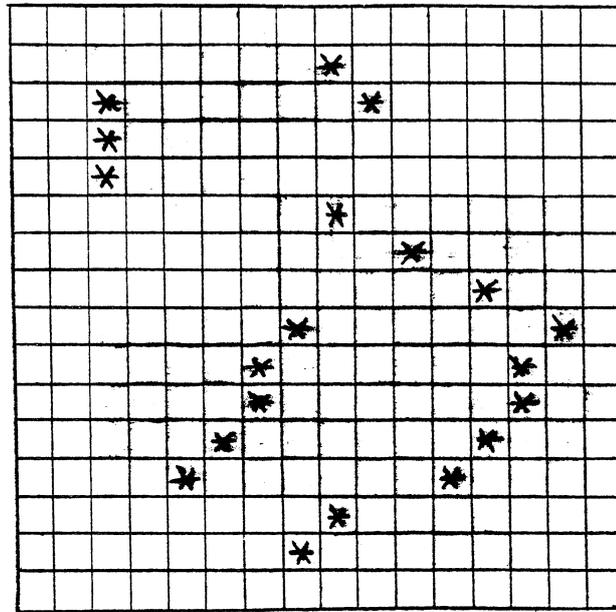
La séparation en régions résultant de ce traitement est montrée à la photo 4.4.

#### 2.1.4 Calcul des équations de plans

A chaque région construite au pas précédent, on associe maintenant une équation de plan. Le calcul des paramètres de cette équation se fait comme au pas 1. La seule différence est que dans le pas 1 on fait un calcul local sur une petite fenêtre alors que maintenant on fait un calcul global sur toute une région.

La procédure de calcul la plus immédiate consiste à donner à l'algorithme d'approximation la liste des pixels  $(X_i, Y_i, Z_i)$  constituant chaque région. Cela est possible, puisque l'analyse de connexité donne cette liste. Toutefois, cette procédure a le défaut suivant: suivant le pas de discrétisation de l'image  $f(X,Z)$ , une même région sera constituée d'un nombre assez variable de points  $(X_i, Y_i, Z_i)$ , ce qui rend le temps de calcul variable lui aussi.

C'est pourquoi nous avons implanté une procédure sensiblement différente qui consiste à prendre un nombre constant  $N$  de points de manière uniforme dans chaque région (nous avons utilisé  $N=20$ ). La sélection de ces pixels est effectuée une fois l'analyse de connexité terminée, alors que l'on connaît le nombre  $NB$  de pixels de chaque région. Nous prenons alors tous les points d'une région dont le numéro d'ordre est un multiple de  $(NB-1)/(N-1)$  dans la liste des pixels fournie par l'analyse de connexité (rappelons que cette liste est construite en parcourant l'image ligne par ligne). Le résultat de cette sélection est illustré par la figure 4.13. Si  $N > NB$ , alors on prend tous les points de la région.



$$nb = 134$$

$$N = 20$$

$$(nb-1)/(N-1) =$$

$$133/19 = 7$$

figure 4.13

Remarque :

Quand on a calculé le meilleur plan au sens des moindres carrés pour chaque région, on est aussi en possession d'un écart quadratique. Si celui-ci est supérieur à un certain seuil  $S$ , on peut conclure que la région n'est pas un plan et que la scène ne comporte pas que des objets polyédriques.

### 2.1.5. Croissance de région

Le pas 3 permet d'affecter une étiquette à un certain nombre de points de l'image  $f(X,Z)$ . Cette étiquette définit la région (portion de plan) à laquelle ils appartiennent. Le seuil utilisé au pas 2 avait été choisi de façon à éviter les "ponts" entre régions. Certains points appartenant effectivement à des régions planes ont donc pu être perdus par cette opération et il s'agit maintenant de les récupérer. Cette récupération est effectuée par un algorithme de croissance de région qui utilise les paramètres des régions calculés aux deux pas précédents (voir photo 4.6).

Après le pas 4, on dispose d'une liste  $P_1, P_2, \dots, P_n$  de portions de plans, définies par des équations analytiques. Pour chaque point  $(X_i, Y_i, Z_i)$  de l'image  $Y=f(X,Z)$  originale, qui n'a pas encore d'étiquette, on peut donc déterminer l'ensemble des  $P_j$  qui sont compatibles avec ce point, suivant la distance de ce point au plan, par un critère tel que :

$$\frac{| Y_i - (a X_i + b Z_i + c) |}{\sqrt{(1 + a^2 + b^2)^{1/2}}} \leq d$$

où  $d$  est une distance seuil.

L'algorithme de croissance de régions nécessite de distinguer plusieurs types d'entités géométriques : portion de plan, arête et sommet. Un point non étiqueté qui n'est compatible qu'avec un seul  $P_j$  est étiqueté comme appartenant à  $P_j$ . Un point compatible avec seulement deux portions de plan  $P_j$  et  $P_k$  est étiqueté comme appartenant à une arête. Un point compatible avec trois portions de plans ou plus est étiqueté comme étant un sommet. De cette façon on agrandit les portions de plan  $P_1, P_2, \dots, P_n$  déjà détectées et on crée de nouvelles régions associées aux arêtes et aux sommets. Ce principe n'est correct que si il n'existe pas deux régions de l'image qui ont la même équation, car cela pourrait amener à confondre une arête avec un sommet par exemple. C'est pourquoi nous détectons au préalable les régions de mêmes équations.

Si l'on revient à notre étiquetage, on voit qu'il nous faut une méthode pour étiqueter les régions "arêtes" et les régions "sommets" de telle sorte que deux pixels compatibles avec les mêmes portions de plans reçoivent la même étiquette. Dans ce but, on construit un vecteur  $V$  à  $n$  composantes ( $n$  est le nombre de portions de plans) pour chaque point d'arête ou de sommet détecté, la  $i$ ème composante de  $V$  étant à 1 si le point est compatible avec  $P_i$ , 0 sinon. Si le vecteur existe déjà, cela signifie qu'on a déjà commencé à construire la région "arête" ou "sommet" à laquelle appartient ce point, et on donne à ce point l'étiquette correspondante (qui est associée au vecteur); si le vecteur n'existe pas encore, une nouvelle région "arête" ou "sommet" est créée et son étiquette est associée au vecteur. Un algorithme de recherche arborescente permet de déterminer l'existence d'un vecteur en un temps de l'ordre de  $\log(q)$ , où  $q$  est le nombre de vecteurs mémorisés.

Remarque:

L'étiquetage des pixels des régions "arêtes" et "sommets" n'est utile qu'à des fins de visualisation graphique (voir photo 4.5). On pourrait se contenter de construire seulement les vecteurs  $V$  définissant les arêtes et les sommets.

L'équation de chaque arête est déterminée par le couple de plans auxquels elle appartient. Les coordonnées d'un sommet sont calculées en résolvant le système de trois équations à trois inconnues défini par la donnée des trois équations de plans qui le définissent; s'il y a plus de trois équations, on résout un système de  $n$  ( $n > 3$ ) équations à trois inconnues, en optimisant par les moindres carrés.

Quand le pas "croissance de région" est terminé, il est possible d'envisager de reexécuter le pas 4, car les régions planes étant plus étendues, les équations de plans peuvent être calculées avec plus de précision (cette itération n'a pas été programmée).

### 2.1.6 Détermination des arêtes occultantes

En sortie du pas précédent on dispose d'un ensemble de portions de plans, caractérisées par leurs équations et reliées entre elles par un ensemble de vecteurs qui définissent les arêtes et les sommets auxquels elles se coupent. Il manque néanmoins à cette structure les arêtes occultantes, c'est-à-dire les arêtes qui ne délimitent qu'une face visible (voir figure 4.14). Le but de ce dernier pas de traitement est de les déterminer.

Pour cela on commence par construire le contour (polygonal) de chaque région correspondant à une portion plane (voir photo 4.7). On détermine ensuite ses sommets en appliquant l'algorithme de découpage récursif du paragraphe 2.1 du chapitre 2. Chaque arête nouvelle (relativement aux arêtes déjà connues) est une arête occultante dont l'équation est déterminée par les coordonnées de ses extrémités. L'intersection de deux arêtes occultantes correspond à un nouveau sommet.

#### Remarque 1 :

Nous avons implanté ce pas 6 pour être complet. Il n'est cependant pas indispensable. Pour construire le modèle d'un objet à partir de plusieurs images  $Y=f(X,Z)$ , on peut très bien avoir des descriptions d'images incomplètes, chaque image ne décrivant, par définition, qu'incomplètement l'objet. Lorsqu'on désire identifier un objet dans une scène comportant plusieurs objets pouvant se recouvrir partiellement, les arêtes occultantes sont souvent peu significatives.

#### Remarque 2 :

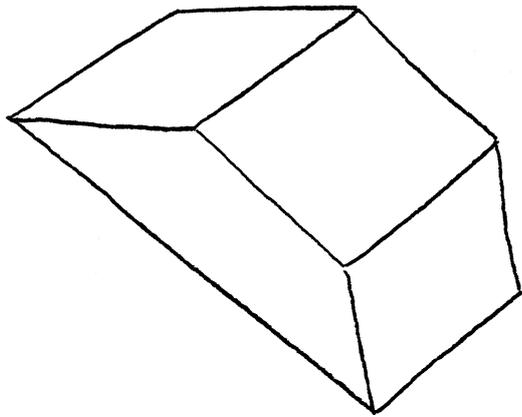
Nous avons implanté un algorithme qui récupère les contours des régions planes en un balayage de l'image étiquetée au pas 5. L'intérêt de cet algorithme est toutefois discutable. En effet, il oblige à traiter toute l'image, alors qu'un traitement région par région pourrait être plus rapide. Il ne se justifie que si l'image est stockée dans une mémoire séquentielle, ou si le pas 6 est exécuté en "pipe-line" avec le pas 5.

### 2.1.7 Autre exemple

Les photos 4.15 (a),(b) et (c) montrent un autre exemple de segmentation.

## 2.2 Cas des quadriques

Dans le paragraphe précédent nous nous sommes intéressé à la segmentation d'images 3D en portions de plans (scènes formées d'objets polyédriques). Nous allons maintenant voir comment les traitements peuvent être modifiés pour les étendre à la détection de portions de quadriques. Nous commençons par le cas où les différentes portions de plans et de quadriques se coupent suivant des arêtes correspondant à une variation d'orientation des plans



objet polyédrique

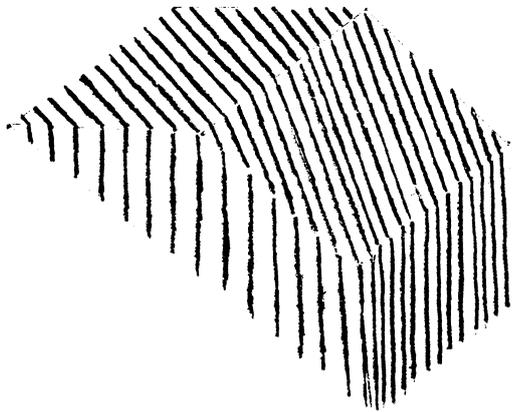
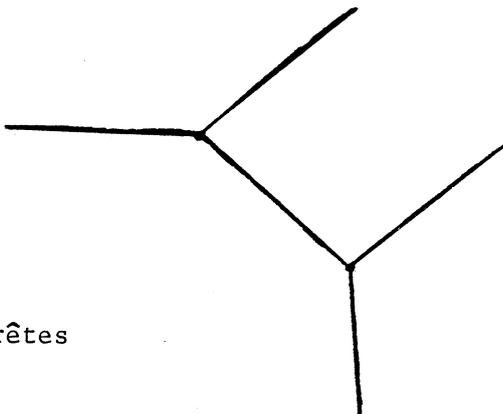
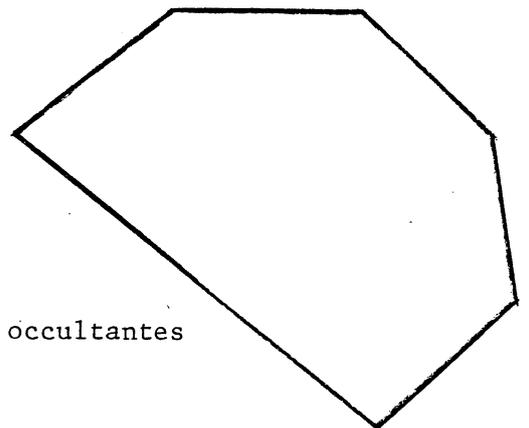


image des lignes



arêtes



arêtes occultantes

figure 4.14

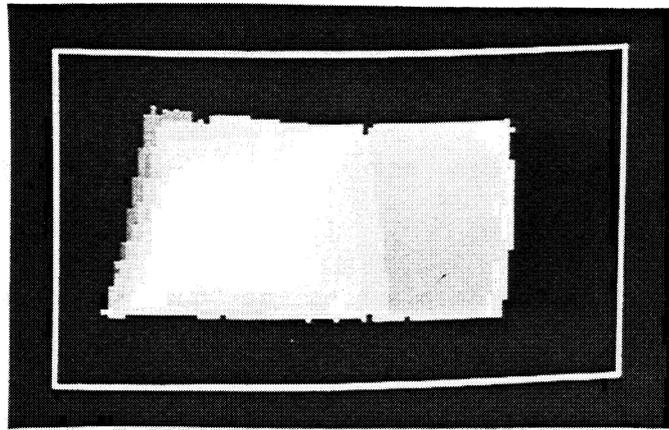


figure 4.15 (a)

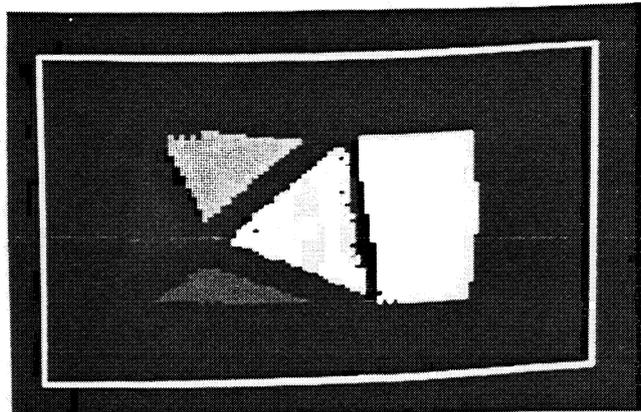


figure 4.15 (b)

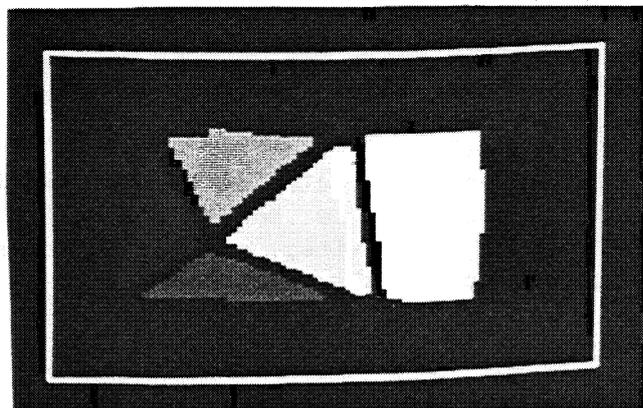


figure 4.15 (c)

tangents. Nous considérons ensuite le cas des portions de plans et de quadriques pouvant avoir les mêmes plans tangents à leur intersection.

Aucune des extensions présentées ci-dessous n'a encore été implantée.

a) Cas de portions de plans et de quadriques se coupant suivant des arêtes

Les trois premiers pas décrits au paragraphe 2.1:

- détection locale des plans
- seuillage adaptatif
- analyse de connexité

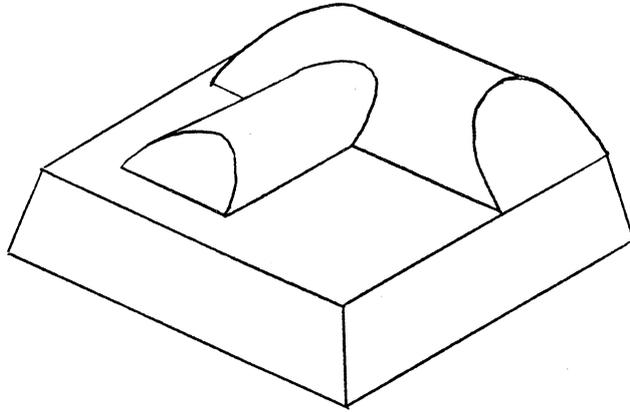
envisagés pour le cas des objets polyédriques conviennent encore et à la fin de l'analyse de connexité, chaque région connexe correspond à une portion de plan ou de quadrique (voir figure 4.16 (b)).

Le quatrième pas calcule le meilleur plan pour chaque région connexe. Si une de ces régions n'est pas un plan (un cylindre par exemple), l'écart quadratique calculé sera trop grand et on saura que la région connexe n'est pas plane. Dans ce cas, on calcule les paramètres de la la meilleure quadrique au sens des moindres carrés qui passe par les points de cette région. Un calcul sur ces paramètres permet de savoir si on a affaire à un cylindre, une sphère, un cône ou un ellipsoïde [Muller-84]. Là encore, il faut se soucier de l'écart quadratique. Si cet écart est au dessus d'un certain seuil, cela signifie que la région connexe n'est pas une portion de quadrique, mais éventuellement plusieurs quadriques non séparées par des arêtes. L'algorithme détecte alors que la scène n'est pas conforme à ce qu'il peut traiter.

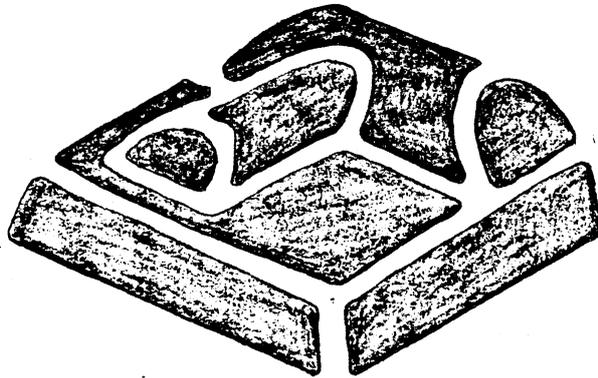
Les régions connexes étant modélisées avec des plans et des quadriques, il reste à modéliser leurs frontières (voir figure 4.16 (c)). Dans le cas des plans, cette modélisation était implicite, puisque la donnée des deux équations de plans est une des spécifications possibles de droite dans l'espace. De plus, deux plans n'ont qu'une seule intersection. Dans le cas des cylindres, c'est plus difficile, car l'intersection de deux cylindres détermine dans certains cas deux courbes fermées. Or dans la scène, il est possible qu'une seule des courbes ne soit présente. Il faut donc procéder d'une autre manière et pour cela, envisageons le pas "croissance de région".

Ce pas va se dérouler de la même manière, c'est-à-dire qu'on va calculer, pour chaque point n'ayant pas été étiqueté pendant l'analyse de connexité, quelles sont les quadriques auxquelles il appartient. Les points appartenant à :

(a)



(b)



(c)

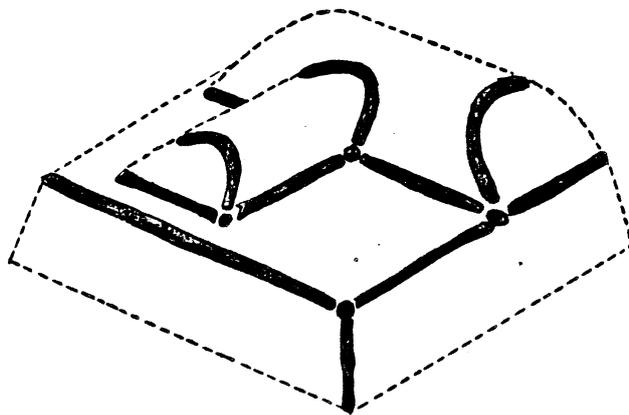


figure 4.16

- une quadrique ou un plan seulement sont rajoutés à la région correspondant à cette quadrique.
- deux quadriques correspondent à des arêtes curvilignes.
- trois quadriques ou plans, ou plus, correspondent à des sommets.

A la fin de ce processus, on dispose pour chaque entité (quadriques, plans, arêtes curvilignes, sommets) des points de l'image 3D originale qui lui correspondent (voir figure 4.16 (b) et (c)).

Pour séparer des points appartenant à la même intersection de quadriques ou plans mais correspondant à des arêtes curvilignes différentes (comme C1 et C2), il faut procéder à une analyse de connexité de ces arêtes. Dans le cas où les arêtes curvilignes sont situées dans un plan, il est possible de calculer l'équation de la meilleure conique au sens des moindres carrés, dans un repère lié au plan en question. Les différentes entités possibles sont les suivantes: droites, paraboles, hyperboles, cercles, ellipses. Dans les autres cas, une description analytique commode des arêtes paraît difficile.

Remarque :

Le pas 6, qui consiste à repérer les arêtes occultantes, a moins d'intérêt dans le cas des quadriques: dans le cas d'un objet polyédrique, les arêtes occultantes correspondent à des arêtes réelles; par contre, avec les quadriques, les arêtes occultantes peuvent ne correspondre à aucune arête curviligne de l'objet.

b) Cas des portions de plans et de quadriques se coupant tangentiellement

Supposons qu'au pas 4 du traitement précédent, l'algorithme ait découvert une région qui ne peut être approchée ni par un plan, ni par une quadrique. On supposera dès lors qu'on est en présence d'un certain nombre de plans et quadriques se coupant tangentiellement, l'absence d'arête n'ayant pas alors permis de les séparer. Il s'agit de détecter ces quadriques pour pouvoir faire une segmentation correcte.

L'approche que nous préconisons pour effectuer une telle segmentation consiste à remplacer le pas 1 ("détection locale des plans") par un pas de "détection locale de quadriques". La fenêtre d'intégration doit être plus grande, pour que la qualité de la quadrique (plan, cylindre, sphère, cône ou ellipsoïde) puisse être correctement appréhendée. L'écart quadratique sera alors plus important sur les frontières entre les quadriques et plus faible sur les quadriques elles-mêmes.

Les pas 2 à 5 restent les mêmes que précédemment.

Il n'est pas certain néanmoins que l'écart quadratique puisse servir de base au seuillage du pas 2. On peut alors envisager d'utiliser les paramètres des plans et quadriques calculés en chaque pixel pour mieux séparer les portions de surface.

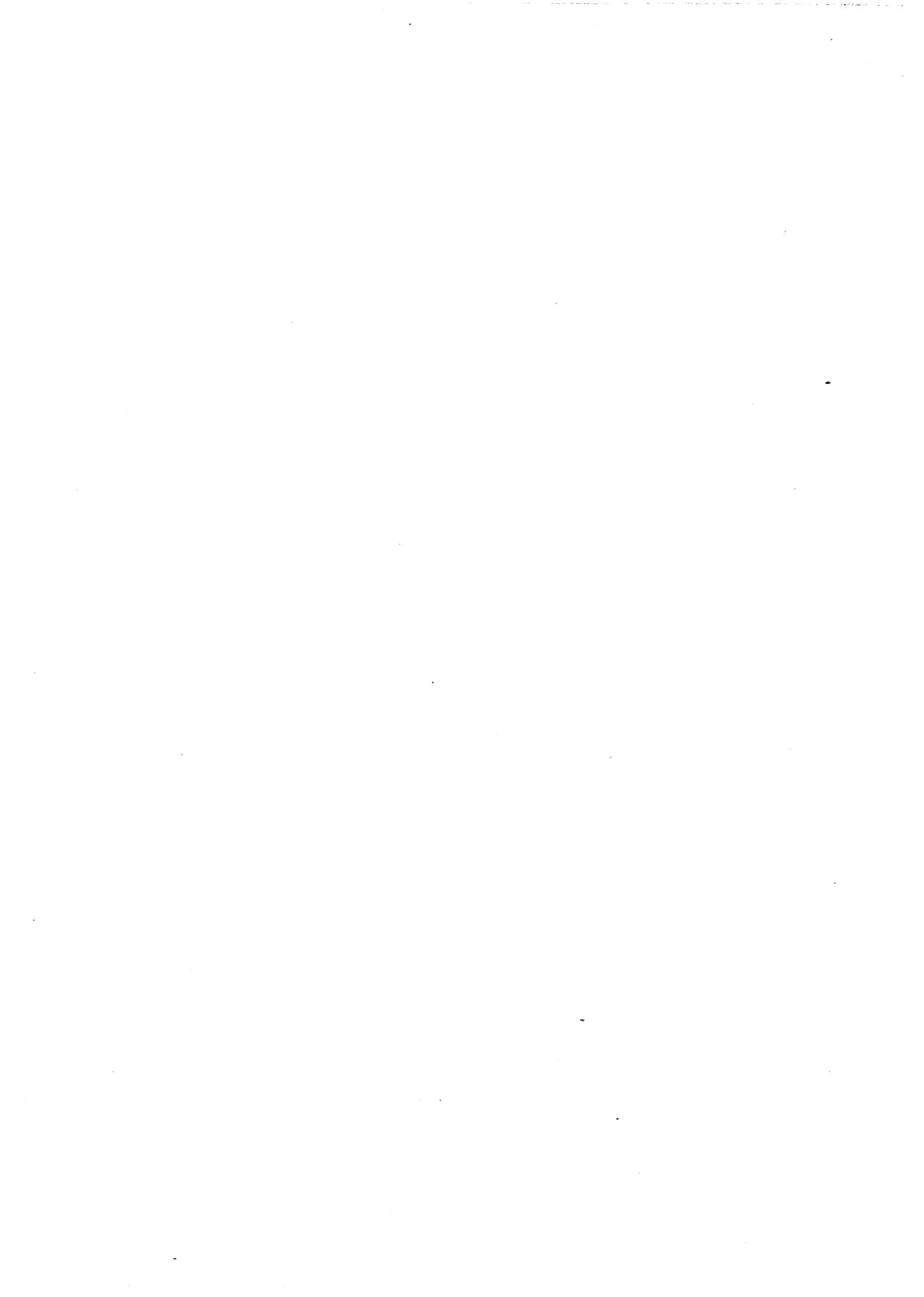
### 3. DISCUSSION

- Des cinq procédés de segmentation présentés, seuls la croissance de régions et le découpage récursif posent à terme un problème réel en temps de calcul, car ils sont d'une complexité nettement non linéaire. La transformée de Hough peut être réalisée de manière parallèle (chaque pixel reçoit le même traitement, indépendamment des résultats des traitements des pixels voisins), mais sans cela elle est coûteuse car chaque pixel nécessite un volume important de calcul. L'analyse de coupes peut être implantée "ligne par ligne". Notre méthode peut être réalisée en mode "pipe-line" (au moins dans le cas de la détection de plans): le premier passage sur l'image sert à calculer l'image des écarts quadratiques et à construire l'histogramme; le second passage réalise le seuillage, l'analyse de connexité et le calcul des paramètres des régions; le troisième passage effectue la croissance des régions et le ramassage de leurs contours; finalement, il reste à segmenter ces contours pour obtenir les arêtes occultantes: ceci doit être réalisé séparément, mais le temps de calcul est assez faible car le nombre de régions est en général assez petit.
- Les algorithmes de croissance de région et de découpage récursif traitent bien les objets polyédriques. Toutefois, il est possible, comme le fait [Shirai-79], après cette facétisation, de recombinaison ces éléments par fusions successives, pour obtenir des portions de quadriques, par exemple. La transformée de Hough permet de détecter en principe toute surface qui peut être décrite par une équation analytique, en particulier les objets polyédriques ou composés de quadriques. L'analyse de coupes peut servir pour les objets polyédriques, plus éventuellement les cylindres. Notre méthode, bien que n'ayant été testée que sur des objets polyédriques, est en théorie applicable à des objets composés de quadriques se coupant non tangentiellement (avec une augmentation des temps de calcul); sa généralisation aux objets composés de quadriques se coupant tangentiellement paraît moins évidente.
- Notre méthode se distingue des autres procédés par l'utilisation de deux niveaux, local (pixel) et global (région). Les pas 1,2,3 et 4 permettent de passer du niveau local au niveau global. Le pas 5 permet un retour au niveau local, pour "récupérer" les pixels non classés dans les régions. Il contribue à la fiabilité de l'algorithme, et à la stabilité des résultats, notamment en les rendant peu sensibles au seuil choisi au pas 2.
- La transformée de Hough est capable de détecter une portion de quadrique, par exemple, mais ne donne aucune indication sur sa localisation. De même deux faces planes distinctes d'un même objet, de même équation, seront confondues. Un post-traitement est donc en général nécessaire. Cela consiste à repérer l'ensemble des points de l'image qui correspondent à l'entité détectée.
- Seul l'algorithme de découpage récursif est sensible aux points "aberrants", les autres introduisant des effets de lissage limitant leur action. Dans notre méthode, ces points sont éliminés, par l'analyse de connexité.



CHAPITRE 5  
CONSTRUCTION AUTOMATIQUE DE  
MODELES D'OBJETS

1. Principe de la mise en correspondance entre deux modèles partiels
2. Génération d'hypothèses
3. Fusion de deux modèles
4. Construction d'un modèle d'objet
5. Application à l'identification/localisation d'objets



Le problème abordé dans ce chapitre est celui de la construction automatique du modèle géométrique d'un objet à l'aide de plusieurs images 3D  $Y=f(X,Z)$  de cet objet prises sous différents points de vue. Sa résolution nécessite un algorithme capable d'engendrer des correspondances partielles entre deux modèles incomplets d'un même objet.

Le paragraphe 1 décrit le principe de l'algorithme de mise en correspondance. Les paragraphes 2 et 3 détaillent l'algorithme. Le paragraphe 4 décrit le procédé de construction de modèles à partir de plusieurs images.

Les applications des algorithmes décrits sont multiples: obtention automatique d'un modèle synthétique d'un objet à priori inconnu, inspection géométrique d'un objet dont on dispose d'un modèle a priori, construction d'une base de modèles d'objets "en montrant" pour un système de vision 3D d'identification/localisation d'objets dans une scène. Cette dernière application nécessite de mettre en oeuvre un algorithme de comparaison entre un modèle d'objet et le contenu d'une image 3D. Nous verrons (paragraphe 5) que l'algorithme de mise en correspondance utilisé pour la construction de modèles peut être réutilisé, éventuellement avec quelques modifications.

Les traitements présentés dans ce chapitre présentent une certaine analogie avec ceux décrits dans [Shirai-81]. Nous expliciterons les points de similitude et de différence.

Les traitements décrits et implantés supposent les objets polyédriques. Leur généralisation à d'autres classes d'objets serait évidemment intéressante. Toutefois, nous ne l'avons pas étudiée, et nous n'en parlerons pas.

Les algorithmes décrits dans ce chapitre ont été implantés dans le langage MACLISP (dialecte de LISP) sur CII-HB-68.

## 1. PRINCIPE DE LA MISE EN CORRESPONDANCE ENTRE DEUX MODELES PARTIELS

Soient  $m_1$  et  $m_2$  deux modèles partiels d'un même objet polyédrique. La mise en correspondance entre ces deux modèles fait appel à deux fonctions principales:

- La première, dite de génération d'hypothèses, tente de mettre en correspondance chaque face de  $m_1$  avec chaque face de  $m_2$ . Le résultat est une liste d'hypothèses de la forme  $(f_1, f_2, T)$ , où  $f_1$  est une face de  $m_1$ ,  $f_2$  est une face de  $m_2$ , et  $T$  est la transformation rigide qui permet de superposer  $f_1$  et  $f_2$  (voir paragraphe 2).
- La seconde, dite de fusion de modèles, engendre pour chaque hypothèse  $(f_1, f_2, T)$  le modèle réunion de  $m_1$  et  $m_2$  suivant  $T$  (voir paragraphe 3).

Un test de cohérence est alors appliqué à chaque nouveau modèle créé.

L'algorithme du principe de la mise en correspondance de  $m_1$  et  $m_2$  est donc le suivant:

```

liste_modèle:=nil;

pour toute hypothèse  $h=(f_1,f_2,T)$  engendrée
    par génération_hypothèse( $m_1,m_2$ ) faire
         $m \leftarrow$  fusion( $m_1,m_2,T$ );
        si cohérent( $m$ ) alors liste_modèle:=cons( $m$ ,liste_modèle);
    finfaire;

```

[Shirai-81], pour faire la correspondance entre un modèle partiel  $S$  associé à la scène et un modèle complet  $M$  déjà enregistré, procède aussi en deux temps que nous détaillerons aux paragraphes 2 et 3. Toutefois la transformation rigide  $T$  qui permet de superposer  $S$  et  $M$  n'est calculée qu'à la fin, à partir de la liste d'association fournie par le processus de correspondance.

## 2. GENERATION D'HYPOTHESES

La complexité de la fonction de génération d'hypothèses est d'ordre  $n_1 \times n_2$ , où  $n_1$  est le nombre de faces de  $m_1$  et  $n_2$  celui de  $m_2$ . Pour réduire le temps d'exécution de la fonction, celle-ci comporte, pour tout couple  $(f_1, f_2)$  de faces de  $m_1$  et  $m_2$  deux pas de traitement:

- Le premier pas consiste à calculer une distance définie sur des propriétés simples des faces (par exemple: aire, nombre de sommets, moment d'inertie), indépendantes de leur contexte (par exemple les relations d'adjacence avec d'autres faces). La comparaison de la distance calculée à un seuil donné conduit à conserver ou à éliminer le couple  $(f_1, f_2)$ . Ce test simple évite, pour de nombreux couples  $(f_1, f_2)$  de se lancer dans le calcul de la transformation qui fait l'objet du second pas.
- Le second pas calcule la transformation rigide (rotationXtranslation) éventuelle  $T$  qui superpose  $f_1$  sur  $f_2$ :  $f_2=T(f_1)$ . Nous le développons ci-dessous.

Le calcul de  $T$  se fait en résolvant par les moindres carrés le système d'équations qui décrit l'association d'une liste  $l_1$  de points de  $f_1$  et d'une liste  $l_2$  de points de  $f_2$ . Pour que le système soit linéaire, les inconnues sont les 12 éléments de la matrice en coordonnées homogènes de  $T$  (9 pour la rotation, 3 pour la translation). Il faut donc au moins quatre points pour chaque liste (chaque point donne trois équations).

La liste de points choisie pour une face se compose des sommets du polygone qui délimite la face, et d'un point dit "normal" obtenu en translatant le centre de gravité de la face suivant la normale extérieure d'une longueur égale à la racine carrée de l'aire de la

face (valeur homogène à une longueur). Avec une telle liste, on évite deux sortes d'écueils:

- Les faces triangulaires ne posent pas de difficulté, car on leur adjoint le quatrième point nécessaire à la détermination des éléments de la matrice qui représente T.
- Des indéterminations sur la valeur de certains éléments de la matrice, dues à la méthode de calcul par les moindres carrés, peuvent subsister lorsqu'on met en relation des listes de points coplanaires.

Une fois les éléments de la matrice calculés, on effectue deux tests:

- Si les 9 éléments correspondant à la rotation de T ne forment pas une matrice 3X3 de rotation (vecteurs non unitaires et/ou non orthogonaux), la transformation T n'est pas rigide et l'hypothèse (f1,f2,T) n'est pas engendrée.
- En calculant les coefficients de la matrice, on calcule aussi l'écart quadratique  $\sum (p_{2i} - T(p_{1i}))^2$ , où  $p_{ji}$  est le  $i$ ème point de la liste  $l_j$ . Si cet écart est supérieur à un seuil donné, l'hypothèse (f1,f2,T) n'est pas engendrée.

Dans tous les autres cas l'hypothèse (f1,f2,T) est engendrée.

La génération d'hypothèse, chez Shirai, revient à n'exécuter que le premier pas de traitement. Shirai recherche dans le modèle partiel S de la scène deux régions  $S_k$  et  $S_k'$  appelées "principale" et "subordonnée" qui vont constituer le "noyau" de départ ("kernel") de son processus de correspondance. La région principale est choisie en fonction de son type ("courbe", "plane" ou "indéfinie"), de sa surface et du nombre des régions adjacentes; la région subordonnée est choisie parmi les régions adjacentes à la région principale, et suivant le même critère. Une hypothèse sera la donnée du quadruplet ( $S_k, S_k', M_k, M_k'$ ) où  $M_k$  et  $M_k'$  sont deux régions du modèle complet M enregistré, satisfaisant à certaines conditions de similarité avec les régions  $S_k$  et  $S_k'$ .

### 3. FUSION DE DEUX MODELES

Soient deux modèles  $m_1$  et  $m_2$ , et une hypothèse (f1,f2,T) fournie par le générateur d'hypothèses pour ces deux modèles. Le but de la fusion de  $m_1$  et  $m_2$  par T est d'engendrer un modèle réunion de  $m_1$  et de  $m_2$ . Le principe de la fusion est le suivant:

- Pour tout couple (s1,s2) de sommets, où  $s_1 \in m_1$  et  $s_2 \in m_2$ , tel que  $s_2 - T(s_1)$  est inférieur à un seuil donné, on fusionne  $s_1$  et  $s_2$ , c'est-à-dire on considère  $s_1$  et  $s_2$  comme un même sommet.
- Pour tout couple (a1,a2) d'arêtes, où  $a_1 \in m_1$  et  $a_2 \in m_2$ , tel que les extrémités de  $a_1$  et  $a_2$  ont été fusionnées, on fusionne  $a_1$  et  $a_2$ .
- Pour tout couple (f1,f2) de faces, où  $f_1 \in m_1$  et  $f_2 \in m_2$ , tel que deux arêtes adjacentes délimitant  $f_1$  ont été fusionnées avec deux arêtes adjacentes délimitant  $f_2$ , on fusionne  $f_1$  et  $f_2$ .

Dans le procédé de fusion que nous avons implanté, le premier pas ci-dessus est itératif. A chaque itération la transformation  $T$  est recalculée en résolvant par les moindres carrés les équations décrivant l'association de tous les sommets fusionnés. Le but de ces itérations est de corriger les imprécisions de la segmentation (chapitre 4). L'expérience montre que la convergence est rapide (en général il n'y a aucune fusion supplémentaire dès la troisième itération). La figure 5.1 montre deux modèles  $m_1$  et  $m_2$ , et le modèle obtenu par fusion de  $m_1$  et  $m_2$  pour l'hypothèse  $(f_1, f_2, T)$ .

Un test de cohérence est appliqué à chaque modèle obtenu par fusion. Dans les programmes implantés ce test consiste à vérifier qu'au plus deux faces se coupent suivant chaque arête. Un test de cohérence plus complet consistera à analyser, pour chaque face, pour chaque arête ou pour chaque sommet, la répartition de la matière alentour (donnée par les directions des normales des faces adjacentes).

Au moment de la génération des hypothèses, Shirai ne calcule pas de transformation rigide. C'est pourquoi la seconde fonction principale de son système n'est pas une fusion entre un modèle  $m_2$  et le modèle  $T(m_1)$  calculé avec  $T$  et  $m_1$ , mais un processus de correspondance entre  $m_1$  et  $m_2$ . Ce processus est incrémental et est initialisé par la donnée de l'hypothèse  $(S_k, S_k', M_k, M_k')$ . Le système essaie d'établir des correspondances entre les régions adjacentes à  $S_k$  et les régions adjacentes à  $M_k$ . Une fonction d'évaluation permet de valider certaines correspondances.

#### 4. CONSTRUCTION D'UN MODELE D'OBJET

Le programme de construction de modèles reçoit en entrée une séquence d'images 3D  $I_1, I_2, \dots$ . Chaque image  $I_i$  est segmentée en une structure  $S_i$  de faces, d'arêtes et de sommets, de laquelle est exclu le fond (face plane constituant le support sur lequel est posé l'objet).

La séquence  $S_1, S_2, \dots, S_i, \dots$  définit une arborescence de modèles:

- $S_1$  est le modèle associé à la racine de l'arborescence;
- soit  $M_i$  ( $i \geq 1$ ) le modèle associé à un sommet de l'arborescence à la profondeur  $i$ ; aux successeurs de  $M_i$  sont associés tous les modèles engendrés par mise en correspondance de  $M_i$  avec  $S_{i+1}$ .

Un sommet de modèle  $M_i$ , à la profondeur  $i$ , n'a pas de successeur si la mise en correspondance de  $M_i$  avec  $S_{i+1}$  ne fournit aucun modèle.

La construction de l'arbre se termine lorsqu'un modèle complet de l'objet a été engendré, ce qui s'exprime par:

$$\text{nombre de faces} - 2 \times \text{nombre de sommets} + \text{nombre d'arêtes} = 2$$

La figure 5.2 montre un exemple d'arbre de modèles. On notera que pour certains objets polyédriques à forte concavité (voir figure 5.3), on ne pourra pas construire un modèle complet. Dans ce cas,

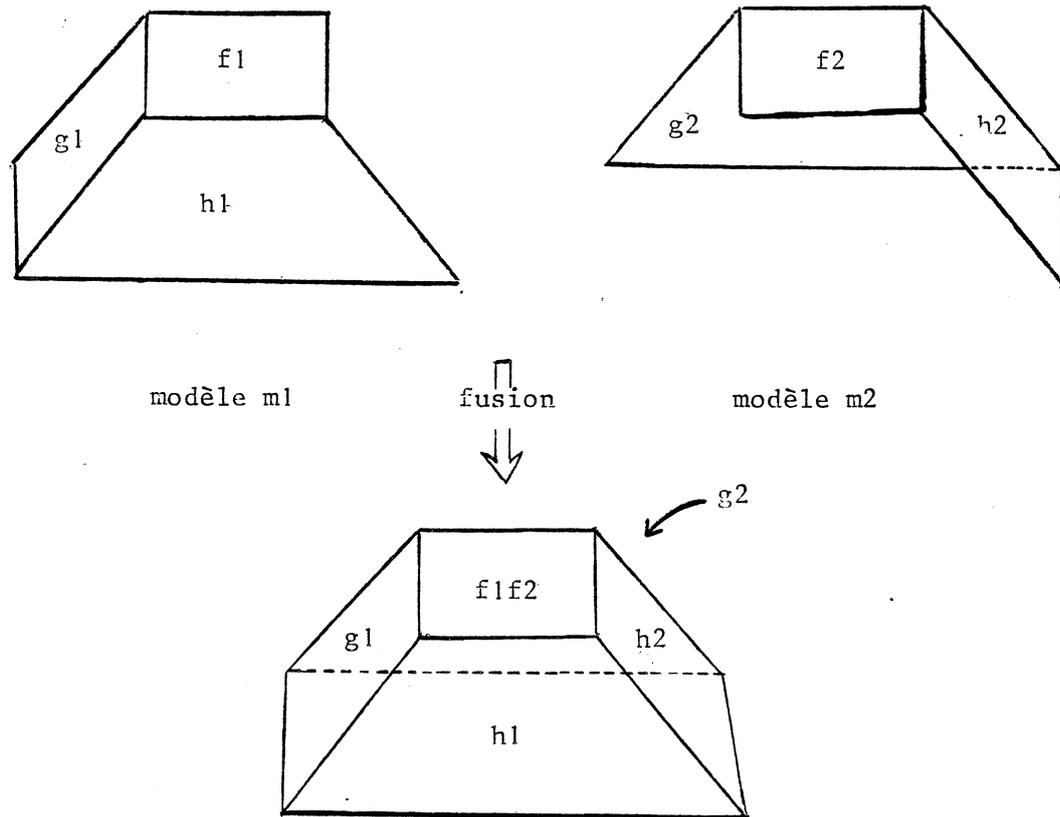
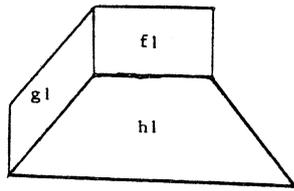
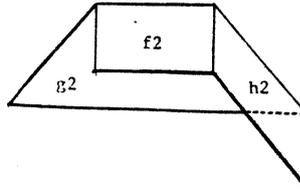


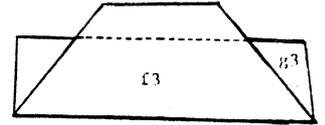
figure 5.1



modèle m1



modèle m2



modèle m3

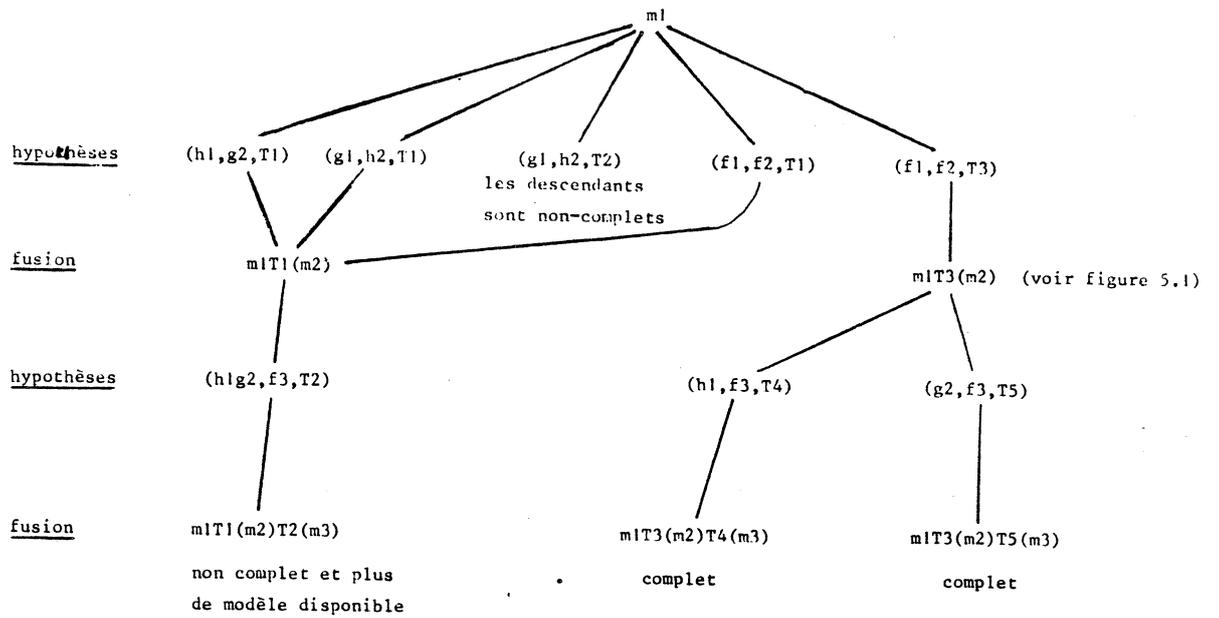


figure 5.2

le système doit engendrer un ensemble de modèles non complets et l'utilisateur doit sélectionner celui qui est correct.

#### 5. APPLICATION A L'IDENTIFICATION/LOCALISATION D'OBJETS

Dans ce paragraphe, nous envisageons l'application de l'algorithme de mise en correspondance décrit aux paragraphes 1,2 et 3 à l'identification/localisation d'un objet dans une scène. On suppose disposer du modèle M de cet objet, éventuellement construit comme il a été décrit au paragraphe 4.

La segmentation de l'image 3D de la scène fournit une structure S de portions de plans, d'arêtes et de sommets. L'identification/localisation de l'objet recherché peut être réalisée directement à l'aide de l'algorithme des paragraphes 1,2 et 3 qui fournira alors des correspondances partielles entre S et M. Un critère simple, tel que le taux de recouvrement de M par les éléments de S intervenant dans chaque correspondance, peut permettre de donner une "plausibilité" à chaque correspondance. Dans ce cas seules les correspondances ayant une plausibilité supérieure à un seuil donné seraient acceptées comme reflétant la présence d'une ou plusieurs occurrences de l'objet recherché dans la scène, la transformation associée à chacune de ces correspondances définissant la position de l'occurrence de l'objet.

Une difficulté liée à l'utilisation de l'algorithme des paragraphes 1 à 3 apparaîtrait si l'objet à identifier ne comporte, dans la scène où il figure, aucune face entièrement visible, du fait de recouvrements partiels. Il serait alors nécessaire de modifier la fonction de génération d'hypothèses (paragraphe 2), fondée sur la correspondance de faces, pour l'étendre à d'autres correspondances (par exemple: paires d'arêtes, triplets de sommets).

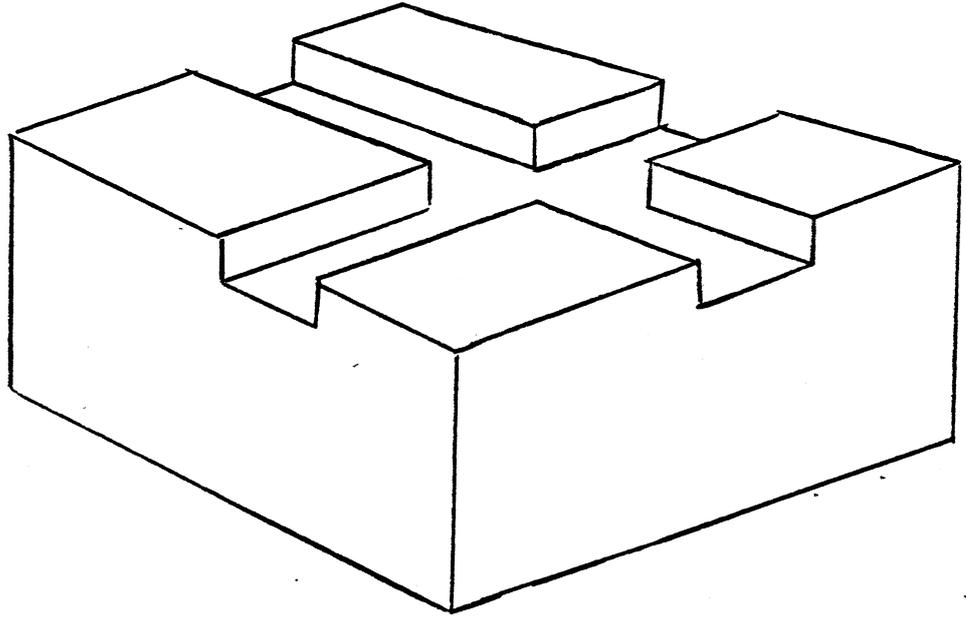


figure 5.3

## CHAPITRE 6

## INTERACTION 2D-3D

1. Intérêt de l'interaction 2D-3D
  - 1.1 Complémentarité des informations 2D et 3D
  - 1.2 Intégration de la forme et de l'aspect visuel
  - 1.3 Aide à la modélisation
2. Calibrage 2D-3D
  - 2.1 Principe
    - 2.1.1 Sens 3D ---> 2D
    - 2.1.2 Sens 2D ---> 3D
  - 2.2 Logiciel de base
    - 2.2.1 Le "quoi"
      - 2.2.1.1 Calibrage espace <---> caméra
      - 2.2.1.2 Primitives diverses
    - 2.2.2 Le "comment"
      - 2.2.2.1 Calibrage espace <---> caméra
      - 2.2.2.2 Primitives diverses



Le capteur 3D que nous avons décrit et utilisé dans les chapitres précédents permet d'obtenir des informations sur la "forme" des surfaces des objets dans une scène. Il ne donne pas d'information sur l'aspect visuel de ces surfaces.

Ce chapitre introduit quelques notions relatives à l'utilisation simultanée du capteur 3D et d'un système de vision 2D, donnant notamment des renseignements sur la luminance.

Nous supposons donc disposer de deux images de la même scène. L'image 2D est notée  $I(i,j)$  et code en chaque point un niveau de gris (intensité lumineuse) et est obtenue avec un éclairage naturel. L'image 3D est notée  $Y(X,Z)$ .

Le paragraphe 1 montre l'intérêt de l'interaction 2D-3D, en supposant connues les relations de calibrage faisant passer des points de l'image 2D aux points de l'image 3D, et vice-versa.

Le paragraphe 2 est consacré à l'établissement de ces relations.

L'ensemble du logiciel décrit dans ce chapitre a été réalisé en FORTRAN sur un ordinateur LSI 11/23. Le système matériel décrit au chapitre 1 (capteur 3D) est ici complété par une caméra avec un numériseur d'images (voir figure 6.1) qui fournit l'image 2D en intensité lumineuse. Cette caméra sera appelée ci-dessous caméra 2D, pour éviter toute confusion avec la caméra du capteur 3D.

## 1. INTERET DE L'INTERACTION 2D-3D

### 1.1 Complémentarité des informations 2D et 3D

D'une façon générale, les informations fournies par deux capteurs visuels, l'un 2D, l'autre 3D, sur une même scène, sont souvent complémentaires. Par exemple:

- Les ombres portées par les objets ne correspondent à aucune entité dans une image 3D, sauf pure coïncidence, alors qu'elles sont délimitées par des lignes de contraste dans l'image 2D (voir figure 6.2).
- Une arête d'un objet peut ne pas faire apparaître une ligne de contraste dans l'image 2D, par exemple si les deux faces qui s'y coupent ont la même luminosité, alors qu'elle est détectable dans l'image 3D (voir figure 6.3).
- Le capteur 3D ne donne aucune information sur l'aspect d'une face, notamment la présence éventuelle d'inscriptions, alors que cet aspect est obtenu à l'aide du capteur 2D.

Une image 2D et une image 3D mettent donc directement en évidence plus d'informations sur une scène qu'une seule d'entre elles. L'interprétation de chaque image peut ainsi être facilitée, à condition de savoir utiliser à chaque instant du processus l'information la plus pertinente. Ceci est réalisé par des stratégies de contrôle appropriées telles que la

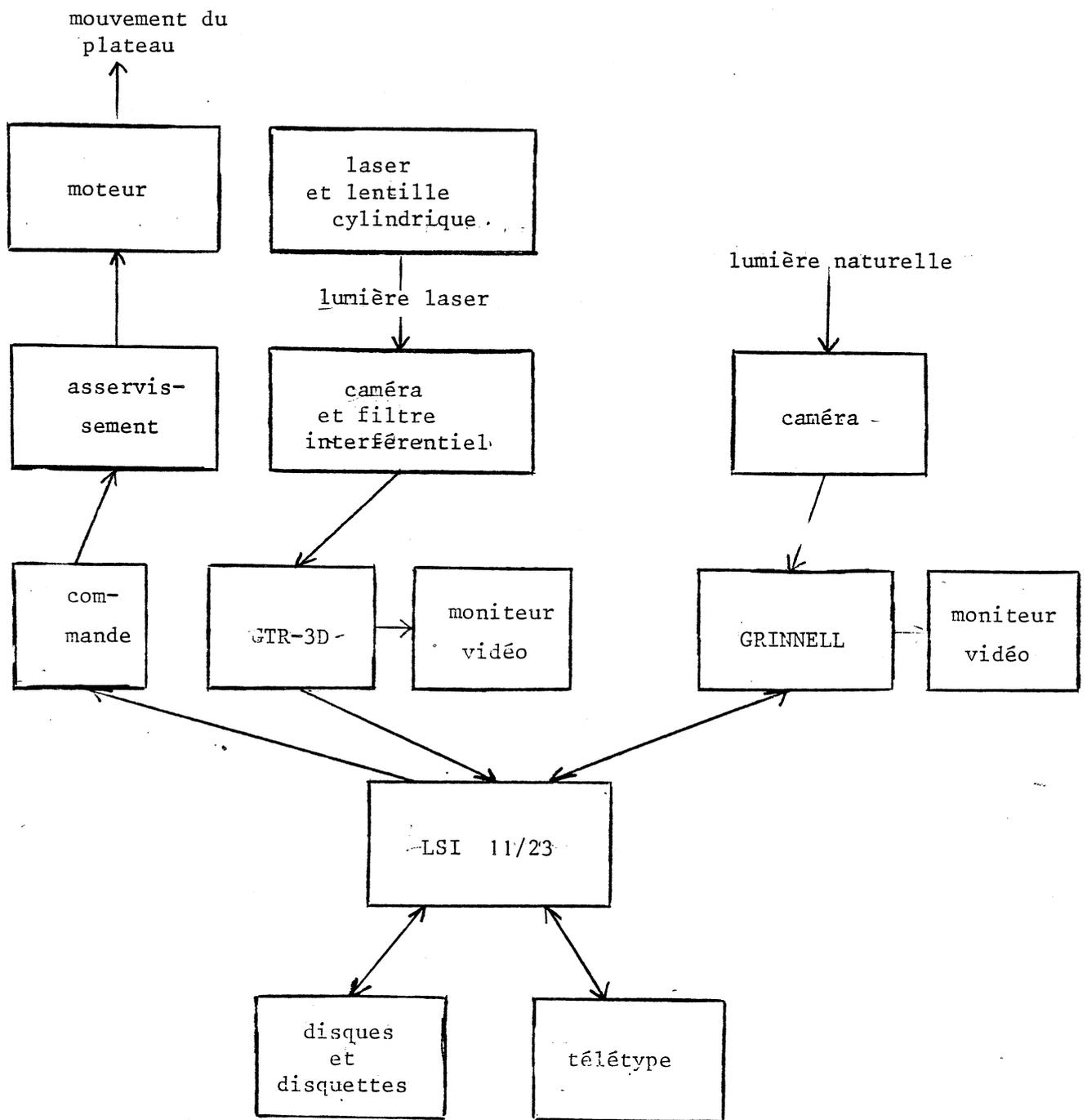


figure 6.1

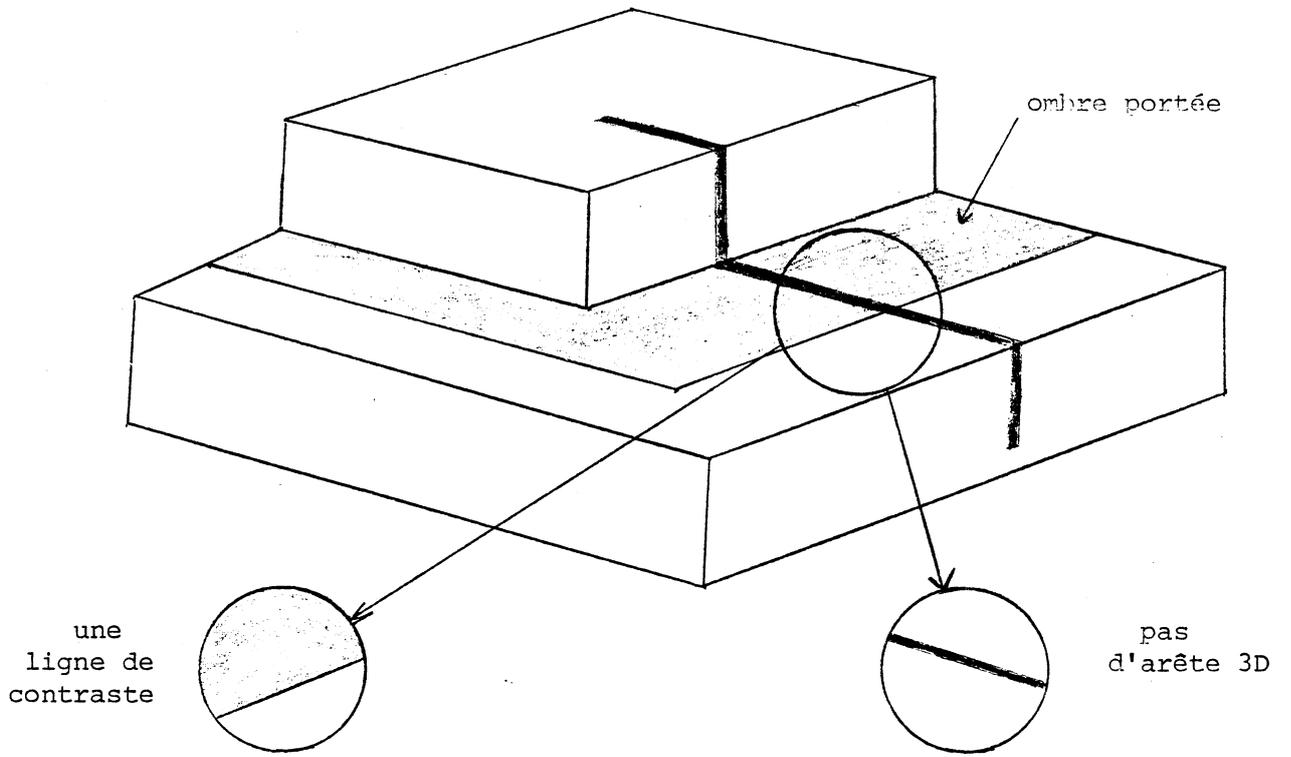


figure 6.2

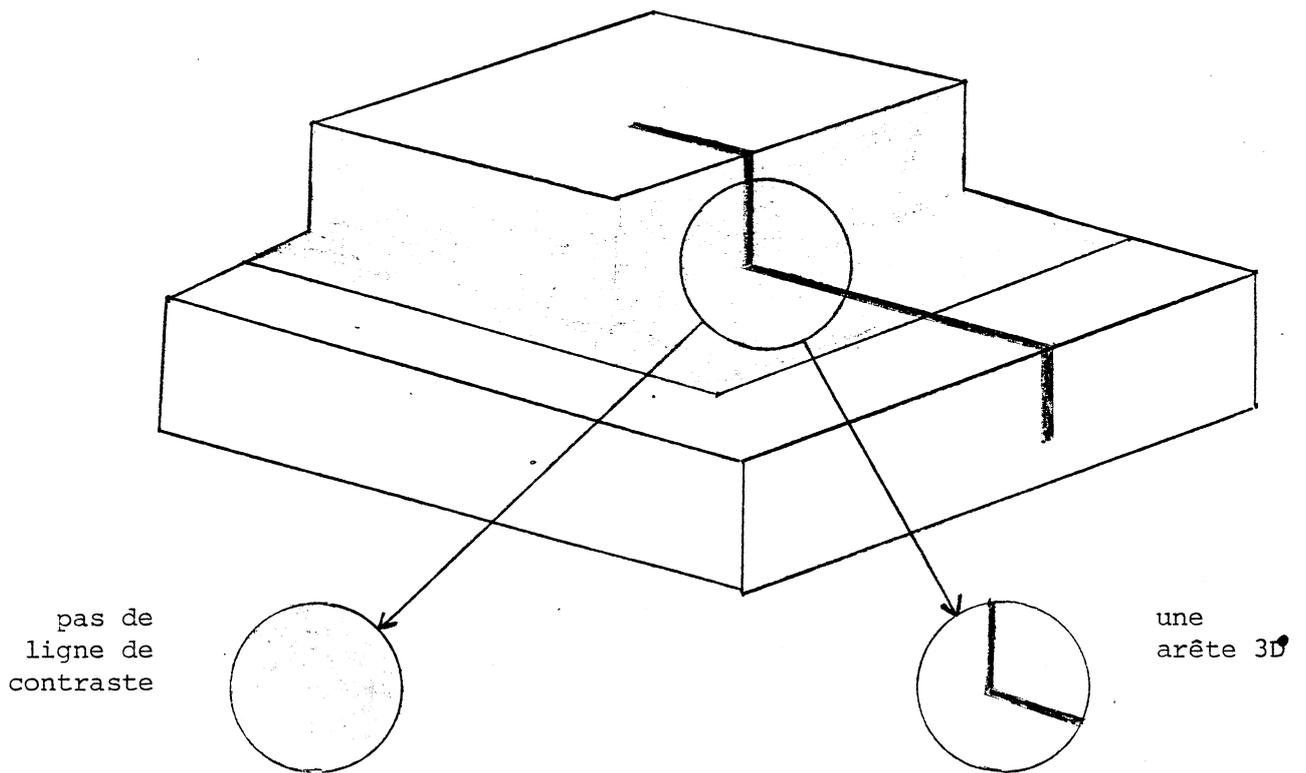


figure 6.3

prédiction-vérification [Souvignier-83] et la génération-exécution de plans [Garvey-76].

### 1.2 Intégration de la forme et de l'aspect visuel

Ce paragraphe prolonge le précédent en intégrant l'image 2D  $I(i,j)$  et l'image 3D  $Y(X,Z)$  en une même image  $S(X,Z)$  donnant en chaque point  $(X,Z)$ , non seulement la coordonnée  $Y$  du point de la surface, mais aussi l'aspect visuel de ce point.

Au point  $M:(X,Y,Z)$  de l'image 3D, se trouve associé un point  $(i,j)$  de l'image 2D, et par voie de conséquence un niveau de gris  $I(i,j)$ . Ce niveau est fonction de l'orientation du plan tangent à la surface au point  $M$  relativement à la source de lumière. Si l'on connaît la position de celle-ci et s'il n'existe pas d'effets d'ombres portées au point  $M$ , on peut calculer la nature intrinsèque (réflectance) de  $M$ . Cela revient à calculer l'angle d'incidence en  $M$  des rayons lumineux relativement au plan tangent.

Si l'image 2D contient des informations de couleur, il est aussi possible d'associer ces informations aux points de l'image 3D.

Disposant de l'image 3D augmentée d'informations de réflectance et de couleur, c'est-à-dire de  $S(X,Z)$ , on peut alors synthétiser des images 2D de la scène suivant différents points de vue et pour différentes positions de la source de lumière [Martinez-82]. Pour cela, on utilise la transformation perspective décrite dans l'annexe D.

### 1.3 Aide à la modélisation

Dans ce paragraphe, nous nous intéressons à l'apport d'une image 2D à la segmentation d'une image 3D construite à l'aide de notre capteur. Cet apport doit être considéré dans le cas où l'image 3D a une faible résolution (par exemple, on a limité le nombre de coupes pour réduire la durée des prises de vue). Nous illustrons cet apport sur deux exemples:

#### - Exemple 1:

Extension d'une arête à l'aide d'une ligne de contraste

Soient  $A$  une arête extraite de l'image 3D (par un procédé quelconque) et  $L$  une ligne de contraste extraite de l'image 2D (par exemple par un système tel que CAIMAN [Lux-81]). Supposons que la mise en relation des deux images (voir paragraphe 2) permette de s'apercevoir que l'image de l'équation de l'arête  $A$  dans l'image 2D est l'équation de la ligne  $L$ . On peut alors calculer le segment 3D  $A'$  qui est supporté par la droite qui supporte  $A$  et qui se projette dans l'image 2D suivant  $L$ . En supposant que  $L$  représente la même arête que  $A$ , on modélisera cette arête par  $AL$ , union des segments  $A$  et  $A'$  (voir figure 6.4).

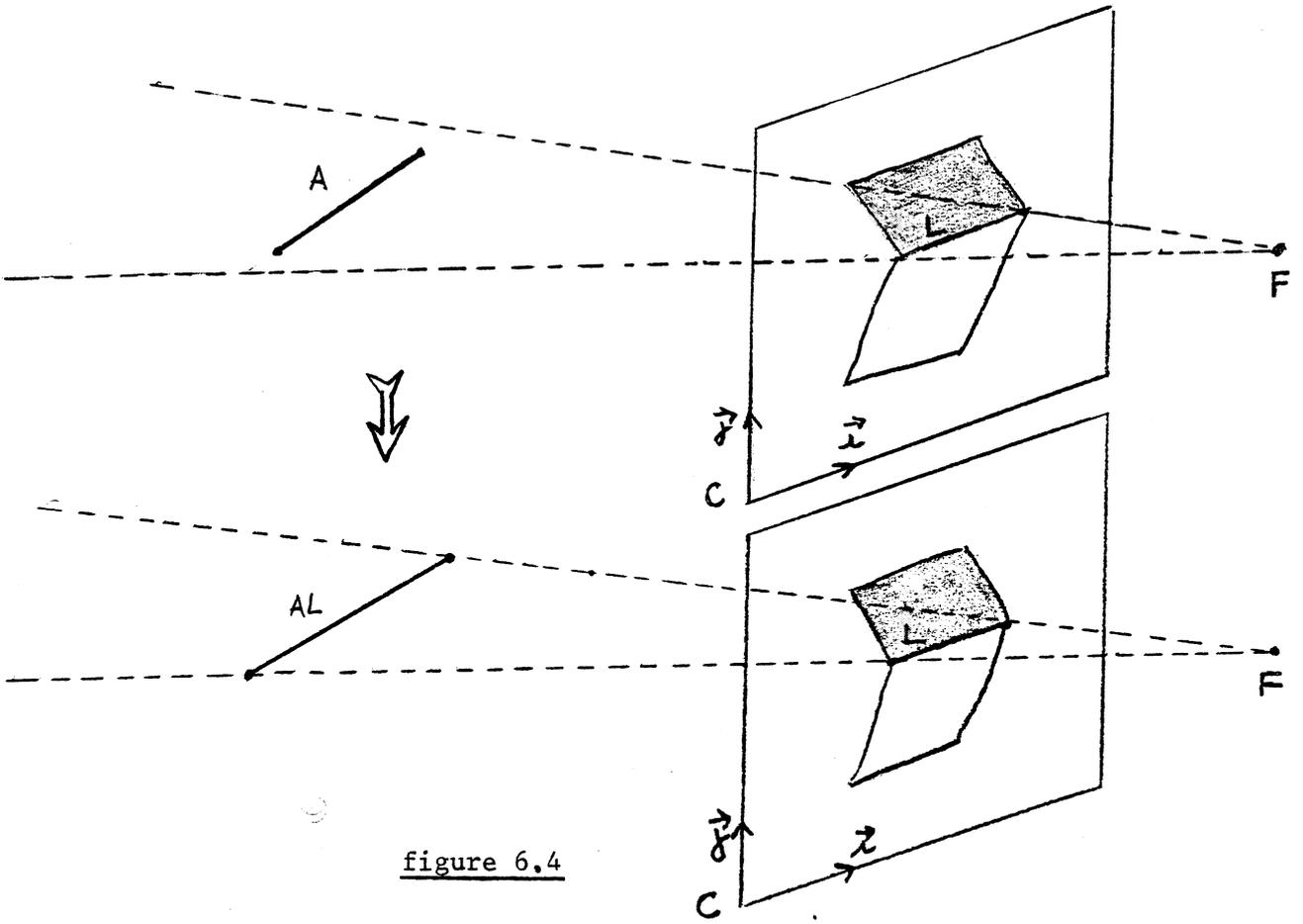


figure 6.4

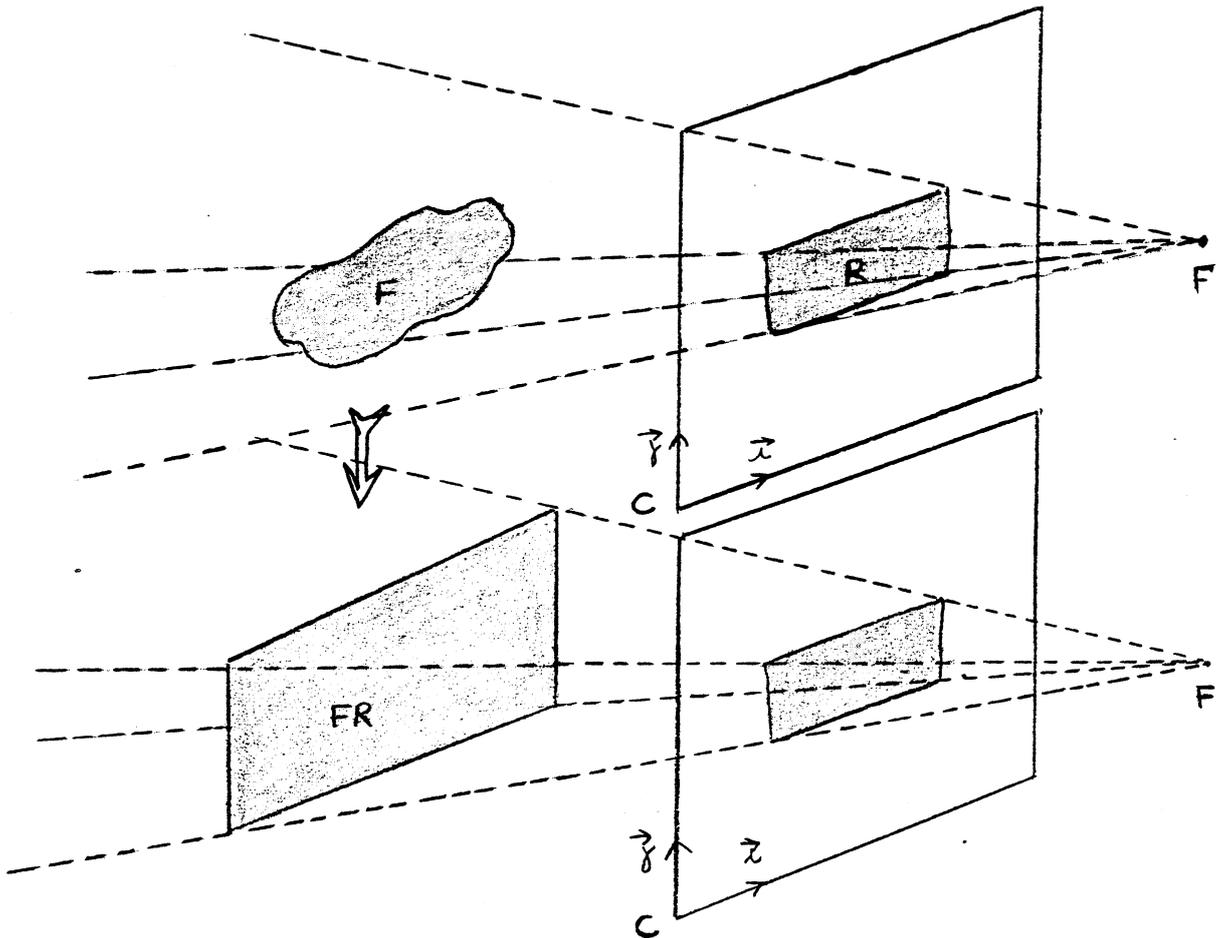


figure 6.5

- Exemple 2:

Extension d'une face 3D à l'aide d'une région 2D

Cette extension repose sur le même principe que précédemment. On met en relation une face F extraite de l'image 3D, et une région R (par exemple d'intensité lumineuse à peu près constante) de l'image 2D. Soit F' la portion de plan qui a la même équation que F et dont l'image par la transformation perspective est R dans l'image 2D. On peut garder comme face l'union FR de F et de F' (voir figure 6.5).

Ces deux exemples montrent qu'il est possible de développer un certain nombre de règles d'interprétation 2D--->3D, qui permettent d'améliorer en précision la segmentation d'une image 3D de faible résolution par l'utilisation d'une image 2D de meilleure résolution. Bien entendu, ces règles sont empiriques et il faut garder un esprit critique quant à leur validité et à leur généralité.

2. CALIBRAGE 2D-3D

2.1 Principe

2.1.1 Sens 3D--->2D

On possède l'image 3D Y(X,Z) et l'image 2D I(i,j) de la même scène.

Le calibrage 3D ---> 2D consiste à déterminer comment un point (X,Y,Z) de l'image 3D est vu par la caméra 2D, c'est-à-dire où, dans l'image 2D, se trouve son image par la transformation perspective.

Les formules suivantes donnent la relation qui existe entre un point de l'espace (X,Y,Z) exprimé dans un repère associé à la scène (repère objet), et son image (i(X,Y,Z),j(X,Y,Z)) dans le repère associé à la caméra (voir annexe D):

$$\begin{cases} i(X,Y,Z) = \frac{a_{11}X + a_{12}Y + a_{13}Z + a_{14}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \\ j(X,Y,Z) = \frac{a_{21}X + a_{22}Y + a_{23}Z + a_{24}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \end{cases}$$

Associer à un point 3D (X,Y,Z) son homologue (i,j) consiste donc simplement à calculer les valeurs i(X,Y,Z) et j(X,Y,Z) en utilisant ces formules. Pour ce faire, il faut connaître les 12 paramètres a<sub>ij</sub> (i=1,4 et j=1,4) de la transformation.

L'opération permettant d'obtenir ces 12 paramètres, appelée calibrage caméra <---> espace, est décrite dans l'annexe D au paragraphe 2. Elle consiste à mettre en correspondance un ensemble d'au moins 6 points M<sub>i</sub> donnés dans le repère objet avec leurs images

Pi dans le repère de la caméra. Chaque association fournit deux équations.

### 2.1.2 Sens 2D--->3D

Il n'est pas possible, connaissant un point de l'image vidéo  $(i,j)$ , de savoir à quel point 3D  $(X,Y,Z)$  de l'espace il correspond, puisque la transformation perspective n'est pas injective, et qu'une infinité de points de l'espace sont vus au même pixel  $(i,j)$ ; ces points sont répartis sur une droite, la droite de visée du pixel  $(i,j)$ , qui passe par ce pixel et le point focal (voir annexe D). Par contre, si l'on sait que le point 3D appartient à l'image 3D :  $Y(X,Z) = f(X,Z)$ , le problème n'est plus le même. En effet, il y a alors, soit aucune, soit une ou plusieurs intersections de la droite de visée du pixel  $(i,j)$  avec la surface  $Y(X,Z)$ . Si la caméra qui fournit l'image 2D est la même (mêmes paramètres internes et externes) que celle utilisée pour faire l'acquisition de la surface 3D, il n'y a au maximum qu'un point d'intersection, dans le cas où c'est non pas l'objet qui se déplace sous le plan lumineux, mais le plan lumineux qui balaye l'objet. Si c'est l'objet qui se déplace (la caméra étant fixe), il n'est pas sûr qu'il n'y ait qu'un seul point.

Le principe du calcul de l'intersection est le suivant: au voisinage du point où la droite de visée du pixel  $(i,j)$  coupe l'image 3D  $Y(X,Z)$ , la droite passe d'un côté de la surface à l'autre. Si on simplifie à une seule dimension, on a (voir figure 6.6):

soit  $D(X-1)-Y(X-1)$  positif et  $D(X+1)-Y(X+1)$  négatif  
soit  $D(X-1)-Y(X-1)$  négatif et  $D(X+1)-Y(X+1)$  positif

ce qui se réduit à  $(D(X-1)-Y(X-1))(D(X+1)-Y(X+1))$  négatif.

En pratique, on procède comme suit. Pour le plan  $Z=tz$ , on calcule l'intersection  $P_1$  de ce plan avec la droite  $D$  de visée du pixel  $(i,j)$  (voir figure 6.7). On calcule la différence  $fly-Ply$  où,  $(Plx,Ply,tz)$  étant les coordonnées de  $P_1$ ,  $fly = f(Plx,tz)$ . On calcule la même différence pour  $tz+1$ . Le couple de plans  $(Z=tz, Z=tz+1)$  qui sera à retenir sera tel que la différence  $fly-Ply$  change de signe. On saura que la droite  $D$  passe d'un côté de la surface 3D à l'autre côté, en passant du plan  $Z=tz$  au plan  $Z=tz+1$ . Le point d'intersection sera calculé comme l'intersection des segments de droites  $P_1P_2$  et  $Q_1Q_2$  (voir figure 6.7).

Si la droite  $D$  coupe l'image 3D en plusieurs endroits, le programme donne la première occurrence rencontrée en parcourant l'axe  $Z$ .

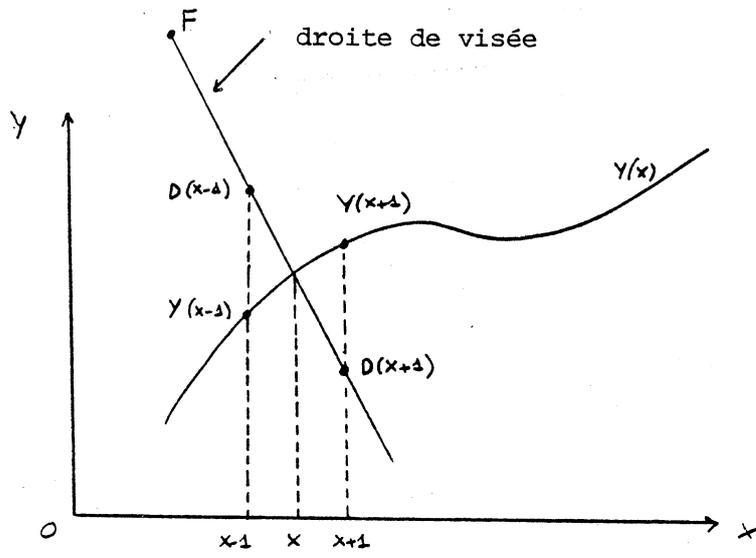


figure 6.6

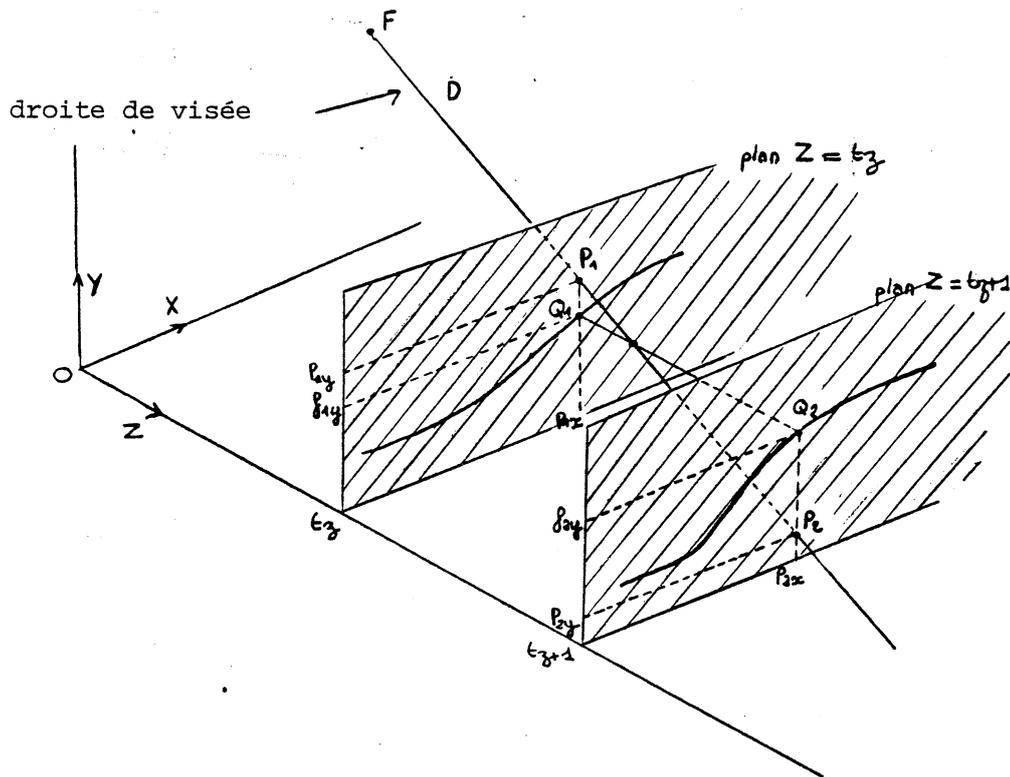


figure 6.7

## 2.2 Logiciel de base

### 2.2.1 Le "quoi"

#### 2.2.1.1 Calibrage espace <---> caméra

Cette fonction détermine les paramètres  $a_{ij}$  de la relation qui lie la position des points de l'espace exprimés dans le repère objet  $(O, X, Y, Z)$  à celle des images de ces points dans le plan rétinien de la caméra.

#### 2.2.1.2 Primitives diverses

##### - Point 3D ---> point 2D

Cette fonction associe à un point espace  $(X, Y, Z)$  son image par la transformation perspective.

##### - Courbe 3D ---> courbe 2D

Soit l'image 3D  $Y(X, Z)$ . On calcule l'ensemble des images des points constituant chaque courbe  $Y(X, tz)$ , par la transformation perspective (voir photo 6.8).

##### - Point 2D ---> point 3D

Cette fonction calcule le point  $(X, Y, Z)$  de l'image 3D  $Y(X, Z)$  qui est vu par la caméra en  $(i, j)$ , cela permet d'associer au point écran, son correspondant dans l'image 3D.

##### - Mire

Cette fonction visualise comment le repère objet  $(O, X, Y, Z)$  est vu par la caméra, en représentant successivement les plans  $X=0$ ,  $Y=0$ ,  $Z=0$ . Cette mire est représentée dans un cadre, qui peut être, d'une manière générale, n'importe quel polygone convexe (voir photo 6.9)

### 2.2.2 Le "comment"

#### 2.2.2.1 Calibrage espace <---> caméra

Comme pour le calibrage du capteur 3D, la réalisation de cette fonction met en oeuvre deux étapes, l'une de traitement d'image, l'autre de calcul numérique. On doit ici repérer non pas quatre, mais six points significatifs dans l'image vidéo. La partie calcul numérique consiste à résoudre un système de 11 (ou plus) équations à 11 inconnues, où les 11 inconnues sont les paramètres  $a_{ij}$ . Cela se réalise facilement avec l'algorithme de Gauss (11 équations) ou des moindres carrés (12 équations ou plus).

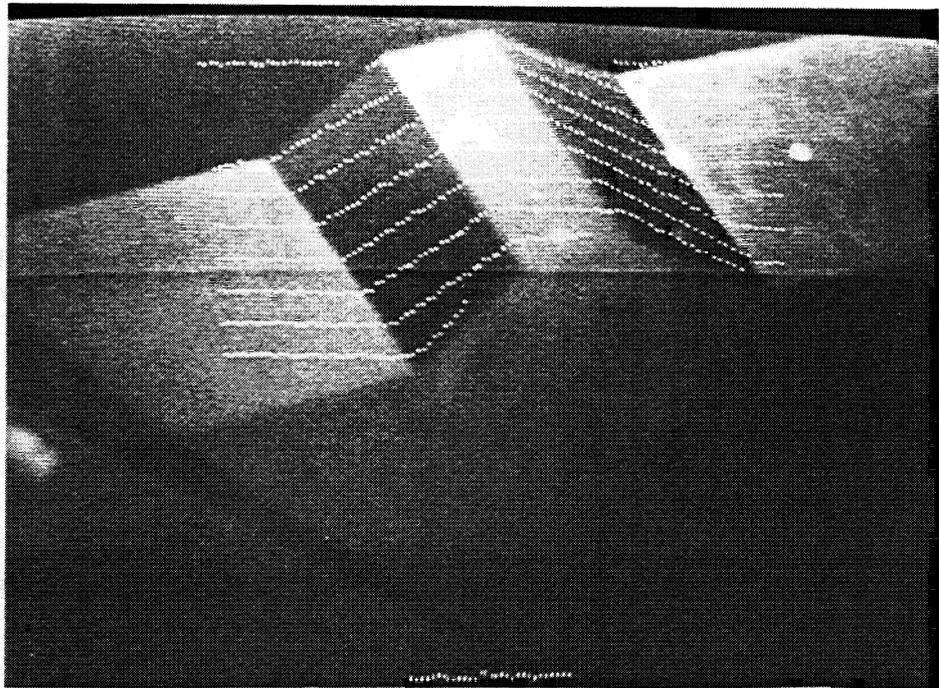


photo 6.8

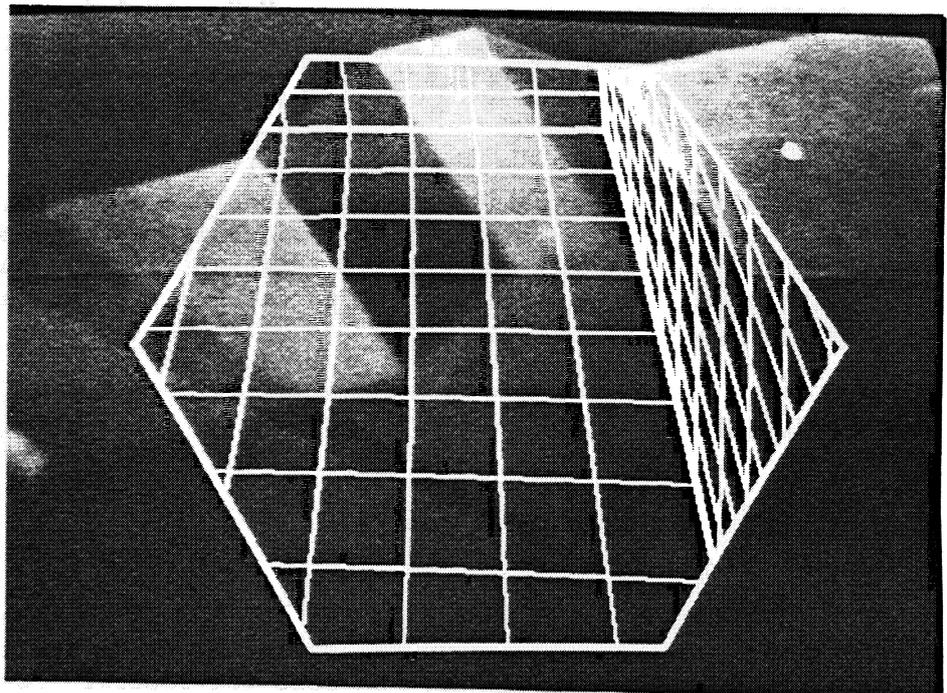


photo 6.9

Le repérage des six (ou plus) points significatifs n'a pas été automatisé. Il est effectué en montrant un objet étalon et en désignant par un réticule sur l'écran les points 2D significatifs.

### 2.2.2.2 Primitives diverses

Pour "Point 3D--->point 2D" et "Courbe 3D--->courbe 2D", voir paragraphe 2.1.

#### - Point 2D ---> point 3D

Si on se reporte à la figure 6.7, on voit qu'il est nécessaire, dans une première phase, de repérer la valeur  $t_z$  qui donne les deux plans  $Z=t_z$  et  $Z=t_z+1$  qui vont servir pour l'interpolation (sur la photo 6.8, on a représenté par des croix les plans sélectionnés, pour le pixel P).

Pour chaque valeur  $t_z$ , on calcule le point d'intersection  $P(t_z)$  entre la droite D et le plan  $Z=t_z$ . Il suffit de résoudre le système (où  $(P_x, P_y, t_z)$  sont les coordonnées de P) :

$$\begin{cases} i = \frac{a_{11}P_x + a_{12}P_y + a_{13}t_z + a_{14}}{a_{31}P_x + a_{32}P_y + a_{33}t_z + a_{34}} \\ j = \frac{a_{21}P_x + a_{22}P_y + a_{23}t_z + a_{24}}{a_{31}P_x + a_{32}P_y + a_{33}t_z + a_{34}} \end{cases}$$

où  $P_x$  et  $P_y$  sont les inconnues.

Le couple de plans ( $Z=t_z$ ,  $Z=t_z+1$ ) qui sera retenu sera tel que la différence  $f(P_x, t_z) - P_y$  change de signe de  $t_z$  à  $t_z+1$ .

Le point 3D sera calculé comme l'intersection des segments  $P_1P_2$  et  $Q_1Q_2$  (voir figure 6.7). Chaque droite donne deux équations dont les inconnues sont les trois coordonnées du point d'intersection. Le problème est résolu facilement en utilisant l'algorithme des moindres carrés pour 4 équations 3 inconnues.

#### - Mire

On visualise de manière graphique le repère objet en représentant successivement les plans  $X=0$ ,  $Y=0$ ,  $Z=0$ . Pour cela, tout repose sur la connaissance des paramètres  $a_{ij}$  obtenus pendant l'opération calibrage caméra<--->espace.

Prenons l'exemple du plan  $X=0$  (pour les autres, cela revient au même). On va tracer successivement les droites  $Y=1$ ,  $Y=2$ ,  $Y=3$ ,... puis  $Z=1$ ,  $Z=2$ ,  $Z=3$ ,... La première série de droites affines correspond à une seule droite vectorielle, c'est-à-dire à une direction de l'espace  $(0,0,1)$ .

On sait qu'à chaque direction de l'espace  $(U,V,W)$  on peut associer un point de fuite  $(P_i, P_j)$  dans le plan rétinien (voir

paragraphe 6 de l'annexe D). Par conséquent, les images des droites  $Y=1, Y=2, Y=3, \dots$  se coupent toutes en un point qu'il est facile de calculer. En effet, si  $(U, V, W)$  est la direction de l'espace en question, le point de fuite  $(P_i, P_j)$  obéit aux équations (voir annexe D):

$$\left\{ \begin{array}{l} P_i = \frac{a_{11}U + a_{12}V + a_{13}W}{a_{31}U + a_{32}V + a_{33}W} \\ P_j = \frac{a_{21}U + a_{22}V + a_{23}W}{a_{31}U + a_{32}V + a_{33}W} \end{array} \right.$$

Donc, pour tracer la droite  $Y=ty$ , il suffit de tracer le segment joignant le point de fuite associé et l'image du point  $(0, ty, 0)$ . Il reste que ce segment, lorsque le point de fuite est très loin, ne peut pas toujours être affiché par le système graphique. Il convient donc de n'afficher que la partie de ce segment qui est contenue dans un cadre prédéfini. Il a donc été jugé intéressant de calculer l'intersection d'un segment avec un polygone convexe (le cadre de visualisation), qui se ramène à un suite d'intersections de segments avec d'autres segments. Deux segments  $P_1P_2$  et  $P_3P_4$  se coupent si:

- $P_3$  et  $P_4$  sont de part et d'autre de la droite affine définie par  $P_1$  et  $P_2$ , et
- $P_1$  et  $P_2$  sont de part et d'autre de la droite affine définie par  $P_3$  et  $P_4$ .

Une fois qu'on sait que l'intersection entre deux segments existe, on calcule l'intersection des deux droites affines en résolvant un système linéaire (2 équations 2 inconnues).

Dans ce mémoire, nous avons présenté successivement le principe et la réalisation matérielle d'un capteur 3D, son logiciel de base, des exemples d'applications, un procédé de segmentation d'images 3D, la construction automatique du modèle d'un objet à partir de plusieurs images 3D, et la coopération d'un système de vision 2D avec le capteur 3D.

Le capteur 3D a de bonnes performances en précision et en vitesse relativement à d'autres capteurs fondés sur le même principe ou sur un principe analogue. Cela est dû au mode de squelettisation du trait lumineux, à l'implantation câblée de cette squelettisation et à la facilité du calibrage. Une extension intéressante du capteur pourrait porter sur l'utilisation de plusieurs caméras et/ou lasers pour aborder le cas d'objets fortement concaves. Un autre extension pourrait consister à avoir plusieurs plans lumineux parallèles simultanément pour accélérer le processus de construction d'une image 3D. Cette seconde extension pose le problème de la séparation des différents traits lumineux perçus simultanément, qui est délicat si les lignes sont discontinues.

Notre logiciel de segmentation d'images 3D peut être amélioré de deux manières. D'une part en ajoutant au logiciel la détection de quadriques et d'autre part en programmant ce logiciel sur un système parallèle avec possibilité d'utilisation en mode "pipe-line", pour augmenter la rapidité du calcul. Nous avons été limité d'autre part par la taille mémoire du calculateur LSI 11/23, ce qui nous a obligé à stocker les informations intermédiaires sur disque, ralentissant le temps de traitement.

En ce qui concerne la construction automatique de modèles, elle peut certainement être améliorée par un choix judicieux des heuristiques guidant la recherche en arbre. Une amélioration importante consiste aussi à incorporer à notre module de correspondance la possibilité de traiter des objets autres que polyédriques, incluant notamment des quadriques.

La coopération de systèmes de vision 2D et 3D n'a été qu'ébauchée au chapitre 6, mais nous pensons qu'elle ouvre des perspectives intéressantes.



- [Agin-79]  
"Sequential strategies for finding and identifying objects with a plane of light" SRI project 4391 & 8487, Machine Intelligence Research Applied to Industrial Automation, 9 th Report
- [Andrieux-83]  
Andrieux Alain, Quénot François  
"La segmentation d'images tridimensionnelles"  
projet de fin d'étude ENSIMAG juin 83
- [Ballard-81]  
"Generalizing the Hough Transform to detect Arbitrary Shapes"  
Pattern Recognition, Vol. 13, nO 2, 1981, p111-122
- [Ballard-82]  
Dana H. Ballard, Christopher M. Brown  
"Computer Vision"
- [Bamba-81]  
Takao Bamba, Hisaichi Maruyama, Eiichi Ohno, Yasunori Shiga  
"A visual sensor for arc-welding robots"  
S.I.R 81
- [Barrow-81]  
H.G. Barrow and J.M. Tenenbaum  
"Interpreting Line Drawings as Three-dimensional Surfaces"  
A.I. vol 17 1981
- [Binford-81]  
"Inferring Surfaces from Images"  
A.I. Vol 17 1981  
AI Lab Stanford Univ., Palo Alto, CA 94305, USA
- [Boissonnat-81]  
J.D Boissonnat and O.D. Faugeras  
"Triangulation of 3D objects"  
Proc 7 th IJCAI pp 658-660 1981
- [Bolles-81]  
"A ransac-based approach to model fitting and its application to finding cylinders in range data"  
IJCAI 81 pp 637-643
- "Three-dimensional locating of industrial parts"  
8 th NSF Grantee's Conf. on Production Research and Technology
- [Bolles-83]  
Robert C. Bolles, Patrice Horaud, Marsha Jo Hannah  
"3DPO : a three-dimensional Part Orientation System"  
8 th IJCAI 83 pp 1116-1120
- [Borienne-81]  
"Capteur 3D" D.E.A. Informatique, LIFIA/IMAG  
Lab. d'I.A. et de Robotique

- [Cain-80]  
"Inspection using planar light"  
Machine Intelligence, Research Applied to Industrial Automation  
SRI project 8487 10 th Report, aug 80
- [Carlbon-78]  
Ingrid Carlbon, Joseph Paciorek  
"Planar Geometric Projections and Viewing Transformation"  
Computer Surveys, Vol 10, No. 4, Dec 78
- [Chehikian-81]  
Chehikian A., Herault J., Mendez J.E.  
"Un processeur temps réel de détection de contours  
dans une image TV"  
Lab. de Traitement d'Images et de reconnaissance de Formes  
ENSERG
- [Comex-81]  
Gérard Mézin, Paul-Louis Borianne  
Rapport de fin de contrat
- [Drouin-83]  
J.J. Drouin, J.P. Braun, P. Paisant  
"Inspection Volumique à l'aide d'un capteur de Vision  
Tridimensionnelle" Projet année spéciale ENSIMAG.
- [Duda-72]  
Duda R.O., Hart P.E.  
"Use the Hough Transform to detect lines and curves in  
pictures" CACM, Vol 15, jan 1972 pp 11,15
- [Duda-73]  
R.O. Duda, P.E. Hart  
"Pattern classification and scene analysis"  
"Perspective Transformation" Chap 10 pp 379-404  
Wiley 1973
- [Duda-79]  
Richard D. Duda, David Nitzan, Phyllis Baret  
"Use of Range and Reflectance Data to find Planar Surface  
Region" IEEE Trans. on Pattern Analysis and Machine  
Intelligence Vol. PAMI-1 No-3 1979
- [Faugeras-82]  
O. Faugeras, F. Germain  
"Vers un système de vision flexible" ,ARA 82  
INRIA Domaine de Voluceau-Rocquencourt BP 105, 78513  
Le Chesnay CEDEX FRANCE
- [Garvey-76]  
"Perceptual strategies for purposive vision"  
A.I. Center S.R.I. Technical Note 117
- [Gelbgras-84]  
Daniel Gelbgras, Marc Bogaert, Gilbert Clermat  
"Robot Control Algorithms for Visual Edge Tracking"  
Advanced Software in Robotics 84 p263

- [Genery-81]  
"A feature-based scene matcher"  
IJCAI 81, Robotics and Teleoperator Group, Jet propulsion  
Laboratory Pasadena, California 91103
- [Grimson-81]  
"From Images to Surfaces"  
A computational study of the Human early visual system
- [Hébert-83]  
"Reconnaissance de formes tridimensionnelles"  
Thèse de troisième cycle
- [Hildreth-80]  
Eller C. Hildreth  
"Implementation of a theory of edge detection"  
AI-TR-579 A.I. lab. MIT
- [Ikeuchi-81]  
Katsushi Ikeuchi and Berthold K.P. Horn  
"Numerical Shape from Shading and Occluding Boundaries"  
A.I. 81 Vol 17
- [Kender-79]  
"Shape from texture"  
Ph.D. Thesis, Department of computer Science,  
Carnegie Mellon Univ., Pittsburgh, 1979
- [Laurgeau-82]  
"Modélisation structuro-paramétrique d'images, Capteur  
autonome en imagerie 3D"  
L.A.N.- ERA 134 NANTES
- [Lelandais-84]  
S. Lelandais, A. Clainchard, J. Lopez-Krahe, H. Maitre, F. Schmitt  
"Numérisation de formes tridimensionnelles : acquisition,  
traitement" 4 ème congrès AFCET 84
- [Lorette-81]  
G. Lorette, M. Gaudaire  
"Méthodes de reconnaissance d'images utilisables en l'absence  
de modèles d'objets et de référentiel unique"  
Univ. PARIS Val-de-Marne, Lab. de Génie-Electrique, Avenue  
du Général de Gaulle, 94010 Créteil cedex, FRANCE
- [Lux-81]  
"Un syst`me de d`veloppement interactif pour le traitement  
d'images" 8 `me colloque sur le traitement du signal et ses  
applications, NICE.
- [Martinez-82]  
"Vers une approche syst`matique de la synth`se d'images- aspects  
logiciels et mat`riels"  
Th`se d`tat, INPG, nov 82
- [Mayhew-81]  
John E.W. Mayhew and John P. Frisby  
"Psychophysical and Computational Studies towards a Theory  
of Human Stereopsis" A.I. Vol 17 1981

- [Mézin-82]  
"Réalisation d'un système d'acquisition d'images tridimensionnelles pour la robotique" Thèse CNAM.
- [Mohr-84]  
"La correspondance en stéréovision vue comme une recherche de chemin optimal"  
Roger Mohr, Brigitte Wrobe  
4<sup>e</sup> me congrès AFCET, 1984
- [Monge-82]  
"Capteur de vision 3D pour la robotique " ,ARA 82  
LETI Lab. Mesure, Contrôle et Traitement Electronique  
C.E.N.G. -85X- 38041 GRENOBLE CEDEX FRANCE
- [Muller-84]  
Muller Yolande, Mirgaux Alain  
"Détection de quadriques par transformée de Hough"  
Congrès AFCET 84
- [Ohta-81]  
Tu-ichi Ohta, Kiyoshi Maenobu, and Toshiyuki Sakai  
"Obtaining surface orientation from texels under perspective transformation" IJCAI 81 pp 746
- [O'Rourke-81]  
"Dynamically quantized spaces for focusing the Hough transform"  
Proceedings of the 7<sup>th</sup> IJCAI, Vancouver 1981 pp 737-739
- [Pavlidis-77]  
"Structural Pattern Recognition"  
Springer-Verlag, New-York 1977
- [Pauchon-83]  
"Numérisation et modélisation automatique d'objets 3D "  
Thèse de docteur-ingénieur, Université de Paris-sud  
centre d'Orsay
- [Popplestone-77]  
R.J. Popplestone A.P. Ambler  
"Forming body models from range data "  
Research Report No: 46 Department of A.I. University of  
Edinburgh
- [Prajoux-82]  
M. Briot, M. Ghallab, A. Izaguirre, RB. Kelley, A. Riad, JC. Talou  
"Système de perception visuelle 2D destiné à identifier  
et localiser des objets partiellement observés"  
ARA 82, LAAS, 7, Avenue du colonel Roche 31400 Toulouse FRANCE
- [Shirai-71]  
Shirai Y. and Suwa  
"Recognition of polyhedrons with a range finder"  
Proc 2<sup>nd</sup> IJCAI p 80-87 London

[Shirai-79]

M. Oshima, Y. Shirai

"A scene description method using three-dimensional information"

Pattern Recognition Vol 11 page 9

[Shirai-81]

Y. Shirai, M. Oshima

"Object recognition using three-dimensional information"

7 th IJCAI pp 601-606

[Sobel-74]

"On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes" A.I. vol 5 No 2

Department of Biological Sciences, Columbia University, New-York, N.Y. 10027, USA

[Souvignier-83]

Viviane Souvignier

"PVV. Un système d'interprétation d'images par prédiction et vérification" Thèse de 3 ième cycle Lab. I.A. et Robotique LIFIA/IMAG

[Sugihara-77]

K. Sugihara, Y. Shirai

"Range data understanding guided by a junction dictionary"

Proc 5 th IJCAI pp 706

[Veillon-77]

F. Veillon

"Une méthode de calcul en un passage de plusieurs propriétés topologiques et géométriques d'objets dans des images digitalisées". Rapport de Recherche no 67.

[Villers-82]

Philippe Villers, President, Automatix inc., USA

"Present Industrial Use of Vision Sensors for Robot guidance"

S.I.R. 82

[Witkin-81]

"Recovering Surface Shape and Orientation from Texture"

A.I. Vol 17 1981 pp 17

[Woodham-78]

"Reflectance Map Techniques for Analysing Surface Defects in Metal Castings" A.I. lab. MIT 78



## ANNEXE A

## DESCRIPTION MATÉRIELLE DU CAPTEUR

## 1. L'ensemble caméra-laser et son électronique de prétraitement

Le laser utilisé est un laser Hélium-Néon MELLES GRIOT de 5 milliwatts émettant une lumière cohérente de 634 nanomètres de longueur d'onde. Sur l'enveloppe protectrice du laser est fixé un barillet pouvant contenir une lentille cylindrique qui transforme le rayon en un plan lumineux, d'une largeur allant de 10 cm à 40 cm, à 40 cm de la lentille cylindrique.

La caméra est une caméra SONY grand public. Son objectif est un zoom Canon J6 X 12, muni d'un filtre interférentiel dont la longueur d'onde est de 634 nanomètres et dont la bande passante est de 7,5 nanomètres (cf. figure A.1).

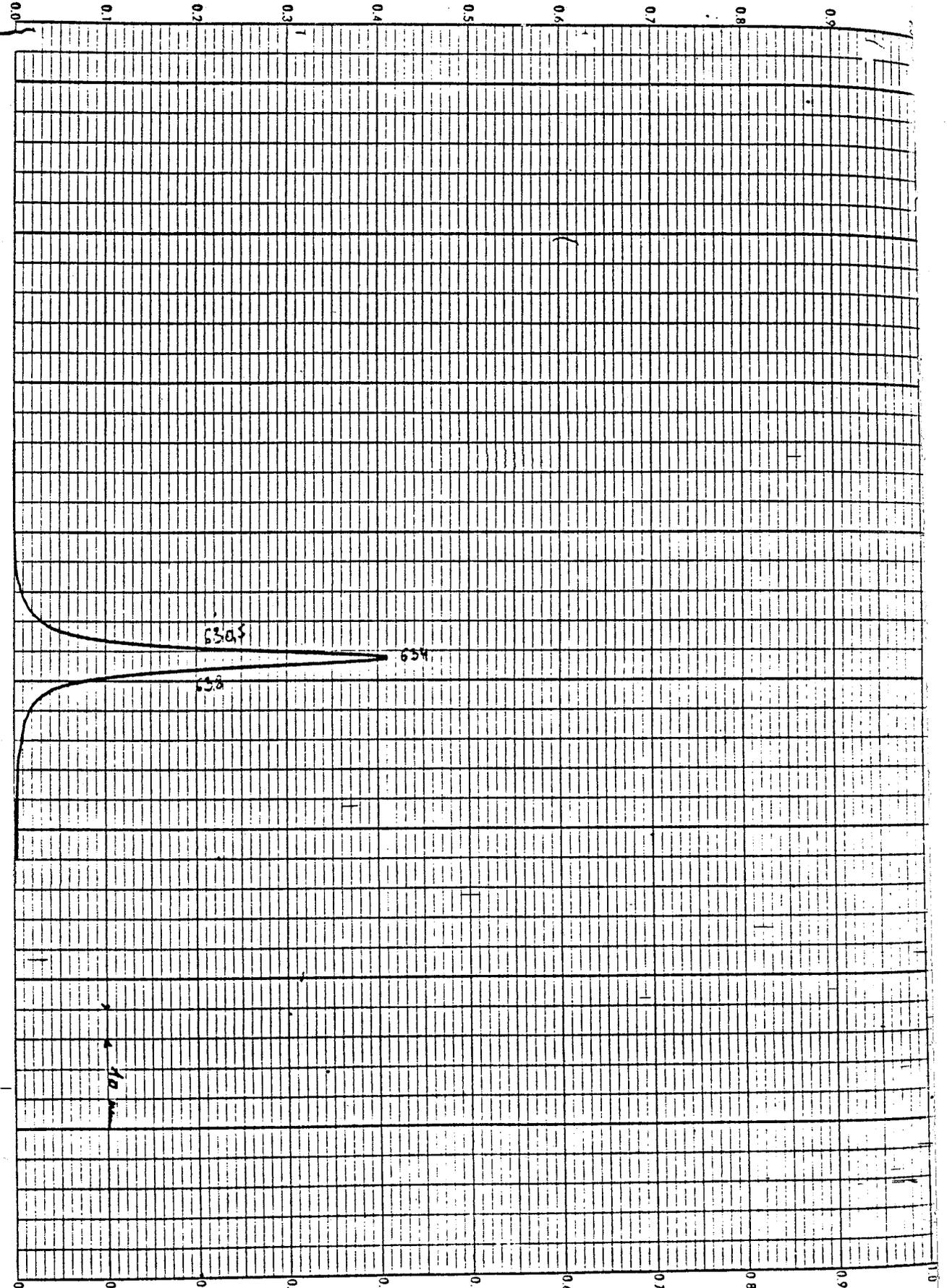
Le signal de la caméra est traité par un processeur ITMI GTR-3D, qui fournit au calculateur le vecteur `CONTOUR[1::256]` qui représente le trait lumineux vu.

## 2. Le plateau translatable

Le plateau est d'une taille de 48 cm X 46 cm. Le moteur utilisé pour assurer la translation est un moteur SIGMA à 400 demi-pas par tour. Compte tenu des démultiplications sur les transmissions il permet la réalisation de déplacements théoriques de 0,02 mm. La vitesse minimale prévue est de 1,5 cm/s et la vitesse maximale de 8 cm/s. La charge maximale prévue sur le plateau est évaluée à 20 Kg.

La commande de puissance sert d'interface entre la commande programmable assurée par un microprocesseur INTEL 8080 et le moteur pas à pas. Elle se compose d'un translateur SIGMA fonctionnant en boucle ouverte. Pour un déplacement à réaliser, le translateur reçoit deux types d'informations: d'une part une donnée binaire qui spécifie le sens de rotation du moteur, et d'autre part un train d'impulsions. Le nombre d'impulsions définit le nombre de pas dont doit tourner le moteur. La fréquence de ces impulsions détermine la vitesse souhaitée pour le moteur.

figure A.1



Examen DA

Rf. 13723  
Indice du support

Incidence : 0°

Dimensions : mm

$\lambda$  : 634 nm.

$\lambda$  : 2.5 nm width.

$\lambda$  : 0.41 nm

Blorngs : *Wittke*

$\lambda$  pour  $T_{min}$  : 1

Dentils oblique

$D_o$  :

$D_i$  :

ADX :

D :

Tracé en :

Transmission

Réflexion

Dentils ordonnés

Echelle ordonnés

Echelle  $\lambda$  :

30 nm inch

Vitesse de balayage :

2 nm/s

Longueur de fente :

Client :

N° de commande :

N° de plan :

N° d'opération :

Tracé effectué sur spectrophotomètre CARY 17 D

Le pilotage du translateur est effectué par une carte BLC 80/10 comportant :

- un microprocesseur INTEL 8080.
- 1K X 8 bits de mémoire RAM.
- 4K X 8 bits de mémoire ROM.
- deux ports d'entrée-sortie parallèles 8255 (48 lignes).
- un port d'entrée-sortie série 8251 pouvant servir d'interface avec un télétype.
- un moniteur (ROM) d'aide à la mise au point.

Le rôle du logiciel implanté sur cette carte est double :

- Il a été conçu pour assurer des fonctions de base dont la réalisation dépend étroitement du système mécanique et de sa motorisation. Il contribue donc à accroître la portabilité du logiciel implanté sur le calculateur LSI 11/23 (cf. paragraphe 3).
- Le microprocesseur 8080 travaille en parallèle avec le calculateur LSI 11/23 et décharge donc celui-ci lorsque plusieurs traitements peuvent être exécutés simultanément. Il est notamment utilisé pour contrôler les déplacements du plateau pendant que le LSI 11/23 de DEC continue le traitement de l'image 3D.

### 3. Le calculateur de traitement

Il s'agit d'un calculateur PLESSEY à base d'un LSI 11/23 dont la taille mémoire est de 64 K mots de 16 bits. Ses périphériques sont deux unités de disques souples structurés en 1000 blocs de 256 mots chacun, plus deux unités de disques durs structurés en 20000 blocks de 256 mots chacun.

Ce calculateur est utilisé avec le système d'exploitation RT11 que nous avons utilisé en mode mono-tache (SJ: single job). Il est connecté au processeur GTR-3D, à la carte BLC 80/10 et au système graphique (cf. paragraphe 4) par des lignes parallèles de 16 bits (cartes DRV11 avec "handshake" permettant d'atteindre des vitesses d'échange de l'ordre de 30 K mots/seconde).

### 4. Le système graphique

Il s'agit d'un matériel GRINNELL GMR 27 construit autour d'une mémoire d'image de 512 X 512 pixels codés sur 4 bits, d'une mémoire graphique de 512 X 512 pixels codés sur 1 bit, et d'un dispositif d'interaction (touches de fonction et manche à balai commandant quatre curseurs indépendants).

Le GRINNEL GMR 27 comporte aussi une entrée vidéo et un numériseur d'image qui a été utilisé pour étudier l'interaction 2D-3D (chapitre 6).

## ANNEXE B

## LES FONCTIONS DU LOGICIEL DE BASE

## 1. Modes d'accès et contenu

Le logiciel de base du capteur est accessible par appel de procédure (appel interne) depuis un logiciel d'application, et à travers un langage de commande interactif simple dont l'interpréteur utilise l'analyseur syntaxique du langage LM réalisé par J.F. Miribel (voir manuel de référence du langage LM). Ce langage est destiné à faciliter la mise en oeuvre du capteur sur de nouvelles applications.

Les fonctions du logiciel de base peuvent être rangées en quatre classes:

- la commande du plateau translatable du capteur seul,
- la prise d'images et la visualisation graphique sans déplacement du plateau,
- la prise d'images et le déplacement combinés,
- divers.

Nous présentons ces fonctions dans les quatre paragraphes ci-dessous. Pour chacune d'elles nous donnons la syntaxe de la commande d'appel interactif, le format d'appel interne, un exemple éventuel et une brève description sémantique. Lorsque certains paramètres n'apparaissent pas dans la syntaxe d'une commande, ceux-ci sont demandés de façon interactive sur le terminal.

## 2. Commande du plateau translatable seul

-1	syntaxe appel interne exemple sémantique	DEPLR <réel> DEPLR DEPLR 3.5 Effectue un déplacement relatif de la position courante p à la position p+3.5cm.
-2	syntaxe appel interne sémantique	CALIB BUTEE CALBUT Précise que la position courante est considérée dorénavant comme la position absolue 0. Dans ces conditions, les butées logicielles sont 0. et 14. (en cm).
-3	syntaxe appel interne	CALIB ARRET ARRCAL

	sémantique	Annule l'opération précédente et considère donc le plateau translatable comme non calibré.
-4	syntaxe appel interne exemple sémantique	DEPLA <réel> DEPLA DEPLA 7 Positionne le plateau translatable à la position absolue 7. Cette opération n'est valide qu'une fois le calibrage butée exécuté (2), et elle n'est plus possible après 3.
-5	syntaxe appel interne sémantique	+ AVNPAS Si le plateau est calibré, équivalent à DEPLA 14, sinon le plateau se déplace vers la droite et s'arrête aussitôt que l'opérateur fait un retour chariot.
-6	syntaxe appel interne sémantique	- RECPAS Si le plateau est calibré, équivalent à DEPLA 0, sinon le plateau se déplace vers la gauche et s'arrête aussitôt que l'opérateur fait un retour chariot.
-7	syntaxe appel interne exemple sémantique	VITESSE <entier entre 1 et 10> VITESSE VITESSE 4 Impose comme vitesse de déplacement la vitesse 4. La vitesse 1 est la plus rapide, la vitesse 10 dix fois plus lente. Par défaut, au début on a la vitesse 4.
-8	syntaxe appel interne sémantique	POSITION POSITN Fournit la position du plateau au moment de l'opération. N'est valide que si le plateau est calibré.
-9	syntaxe appel interne sémantique	ARRET ARRET Interrompt tout déplacement. Si le plateau est calibré, la position d'arrêt est connue.

### 3. Prise d'images et visualisation sans déplacement

-10	syntaxe appel interne sémantique	ECRAN ECRAN Affiche par l'intermédiaire du système GRINNELL GMR 27 une palette de niveaux de gris permettant d'ajuster la luminosité du moniteur vidéo.
-----	--	---

- 11            syntaxe            EFFI  
                 appel interne        EFFI  
                 sémantique            Efface la mémoire d'image du GMR 27.
- 12            syntaxe            EFFG  
                 appel interne        EFFG  
                 sémantique            Efface la mémoire graphique du GMR 27.
- 13            syntaxe            CAMERA  
                 appel interne        CAMERA  
                 sémantique            Met le GMR 27 en prise d'image. L'image s'arrête aussitôt que l'opérateur fait un retour chariot.
- 14            syntaxe            ECRIT ECRAN  
                 appel interne        FICB2  
                 sémantique            Permet de sauvegarder une image du GMR 27 sur disquette ou disque. Le système demande un nom de fichier RT11 puis envoie un message relatif à la partie de l'image qui est sauvegardée. Pour toute l'image, taper "/" et retour chariot. Le suffixe par défaut du nom de fichier est .IMA.
- 15            syntaxe            LIT ECRAN  
                 appel interne        ECRIB2  
                 sémantique            Effectue la même opération que précédemment, mais en sens inverse, c'est-à-dire qu'on visualise une image sauvegardée. Mêmes conventions.
- 16            syntaxe            CALIB CAMERA  
                 appel interne        CALCAM  
                 sémantique            Effectue le calibrage "espace" de la caméra deux qui fournit les images en niveau de gris. Après cette opération la partie "capteur 2D" est valide (pour pouvoir permettre l'interaction 2D-3D).
- 17            syntaxe            MIRE  
                 appel interne        MIRE2  
                 sémantique            Dessine en graphique la mire, c'est-à-dire comment le repère espace est vu par la caméra deux. Le système demande la maille, qui correspond à l'espacement du réseau affiché. En répondant par "oui" ou par "non" aux questions que pose le système, on a la possibilité d'afficher à volonté les plans X=0, Y=0, Z=0 de la mire. Cette opération n'est valide que si 16 a été effectué.
- 18            syntaxe            ECRAN ESPACE  
                 appel interne        ECES  
                 sémantique            Pour un objet donné, c'est-à-dire un ensemble de courbes, et une vue de cet objet, cette primitive fait passer d'un point écran (2D) à un point espace (3D). Le point écran est désigné à l'aide du curseur 1 du GMR 27 et n'est valide que si 16 a été effectuée.

- 19      syntaxe            ESPACE ECRAN POINT  
           appel interne    VISUPT  
           sémantique       Pour un point de l'espace (3D), visualise où  
                                  ce point est vu par la caméra, dans l'écran  
                                  (2D) et n'est valide que si 16 a été effectué.
- 20      syntaxe            ESPACE ECRAN COURBE  
           appel interne    VISUCB  
           sémantique       Visualise un objet 3D, c'est-à-dire un ensemble  
                                  de courbes, tel qu'il est vu par la caméra, d  
                                  l'écran (2D) et n'est valide que si 16 a été  
                                  effectué.
- 21      syntaxe            LIT  
           appel interne    LITDEM  
           sémantique       Visualise un objet 3D, comme un ensemble de  
                                  courbes, à l'échelle désirée.
- 22      syntaxe            GRIS  
           appel interne    GRIS  
           sémantique       Visualise un objet 3D, en représentant  
                                  en niveaux de gris l'altitude en chaque  
                                  point.
- 23      syntaxe            ECRIT  
           appel interne    ECRITC  
           sémantique       Permet de visualiser sur le terminal un objet  
                                  3D, en affichant les courbes à la demande,  
                                  ainsi que divers paramètres liés au codage de  
                                  courbes ( X, Z, etc...).
- 24      syntaxe            FOND <réel>  
           appel interne    FOND  
           sémantique       Donne au système une valeur réelle qui  
                                  correspond à la position du fond. Cette valeur  
                                  est utilisée par la primitive suivante. Cette  
                                  valeur est 0 par défaut.
- 25      syntaxe            ENLEVE  
           appel interne    ENLEVE  
           sémantique       Si p est la profondeur du fond donnée par  
                                  24, enlève tous les points qui ont une  
                                  profondeur inférieure à cette valeur.
- 26      syntaxe            TROU  
           appel interne    TROU  
           sémantique       Remplit les zones non-définies par interpolat  
                                  sur les lignes.
- 27      syntaxe            DYNAMIQUE  
           appel interne    MOYEN2  
           sémantique       Recentre la dynamique d'un objet 3D, pour  
                                  l'affichage en niveaux de gris, à partir de  
                                  la mesure de la moyenne et de l'écart type.
- 28      syntaxe            INVERSE  
           appel interne    INVERS  
           sémantique       Inverse simplement la dynamique dans l'affich  
                                  en niveaux de gris, le clair se trouve à la p

du foncé, et vice-versa.

- |     |  |  |
|-----|--|--|
| -29 | syntaxe<br>appel interne<br>sémantique | HISTOG<br>HISTO2<br>Calcule un histogramme sur l'image 3D de la fonction $Y=f(X,Z)$ , et l'affiche sur l'écran.  |
| -30 | syntaxe<br>appel interne<br>sémantique | CALIB PRISE<br>CALPRI<br>Met le GMR 27 en prise d'image et affiche le ca de prise de vue: cela permet de positionner la caméra pour que les objets à mesurer soient visibles (dans le cas du capteur fonctionnant sans GTR 3D, c'est-à-dire tout en logiciel). |
| -31 | syntaxe<br>appel interne<br>sémantique | PRISE<br>PRISE<br>Effectue une prise de vue à la position courant à partir du moment où l'opérateur tape un retour chariot qui arrête la prise d'image.  |

#### 4. Prise d'image et déplacement combinés

- |     |  |  |
|-----|--|--|
| -32 | syntaxe<br>appel interne<br>sémantique                               | CALIB VERT<br>CALVER<br>Le plateau se déplace jusqu'à la position donnée par 34 et le système optique s'autocalibre.   |
| -33 | syntaxe<br><br>appel interne<br>exemple 1<br>exemple 2<br>sémantique | PRISEA <entier><réel><réel> ECRIT<br>(optionnel)<br>PRISEA<br>PRISEA 6 1 5<br>PRISEA 20 5.3 2.1 ECRIT<br>Dans le cas de l'exemple 1, le système fait 6 prises de vue entre les positions absolues 1.0 et 5.0, bornes comprises et n'est valide que si le plateau translatable est calibré (2). Pour l'exemple 2, on fait 20 prises de vue entre les positions 5.3 et 2.1 et les courbes sont stockées sur disquette ou disque au fur et à mesure. Il est demandé un nom de fichier RT11 dont le suffixe par défaut est .IMA. |

#### 5. Fonctions diverses

- |     |   |  |
|-----|---|--|
| -34 | syntaxe<br>appel interne<br>exemple<br>sémantique | CALIB POSITION <réel><br>CALPOS<br>CALIB POSITION 14<br>Pour l'opération 32, calibrage vertical, le plateau se positionnera en 14. |
| -35 | syntaxe<br>appel interne                          | CALIB ?<br>CALDEF  |

- sémantique Indique quels sont les calibrages effectués:  
zéro: non effectué, un: effectué.
- 36 syntaxe FIGE  
appel interne FIGE  
sémantique Ecrit sur disquette ou sur disque un fichier  
contient tous les paramètres du système au  
moment de l'opération. C'est une sauvegarde.
- 37 syntaxe DEFIGE  
appel interne DEFIGE  
sémantique Même chose que précédemment, mais dans l'aut  
sens. C'est une restauration.
- 38 syntaxe FICHIER  
appel interne FICHIER  
sémantique Désigne pour tous les utilitaires quelle est  
l'image 3D à utiliser (c'est un nom de fichier  
RT11).
- 39 syntaxe FORMAT <entier entre 1 et 10>  
(optionnel)  
appel interne FORMAT  
sémantique Si une valeur est donnée, affiche l'image  
3D en niveau de gris (22) suivant un des modes  
préprogrammés. Sinon, cela permet de définir  
mode d'affichage (paramètres de translation  
d'homothétie, et courbes sélectionnées).
- 40 syntaxe FIN  
appel interne FINISH  
sémantique Sort du logiciel V3D et rend la main au moni  
RT11.

## ANNEXE C

## LOGICIEL DE MOUVEMENT DU PLATEAU

Il s'agit du logiciel de base permettant la commande par le calculateur LSI 11/23 du plateau translatable. Une partie de ce logiciel se trouve sur le calculateur maître (LSI 11/23) et l'autre partie se trouve sur la carte BLC 80/10 qui possède un microprocesseur INTEL 8080 et des possibilités d'entrées-sorties.

## 1. Matériel utilisé pour l'échange et principe

La carte utilisée pour réaliser l'échange est une interface parallèle 16 bits DRV11, du côté du LSI 11/23, et des ports d'entrées-sorties (8255) avec possibilité de signaux de contrôle du côté du micro INTEL 8080. Les registres qui permettent l'échange, pour la carte DRV11, sont adressés comme des éléments mémoires dont l'adresse est spécifiée par des cavaliers ou des interrupteurs sur la carte. Ces registres sont au nombre de trois :

- RC registre de contrôle
- RS registre de sortie
- RE registre d'entrée

En testant certains bits du registre de contrôle RC, on a la possibilité d'effectuer un échange avec "handshake". Les signaux de contrôle sont les suivants (voir figures C.1 et C.2) :

DRV11            =====>            8255

\* New Data Ready



Ce signal à impulsion (Strobe) est envoyé au 8255 par la carte DRV11, aussitôt que le registre de sortie RS est rempli.

\* Req A



Quand le signal New Data Ready, qui signifie que le LSI 11/23 est prêt à envoyer une donnée, est transmis, il met à 1 la bascule IBF (figure C.1 : a) qui permet au 8080 de savoir qu'une nouvelle donnée en entrée est prête. Ce signal est renvoyé à la DRV11 (Req A) et permet au LSI 11/23 de savoir que sa donnée n'a pas encore été lue. Au moment où le 8080 lit son registre d'entrée, il met à zéro la bascule IBF (figure C.1 : b). Ce signal est renvoyé à la DRV11 (Req A) et permet au LSI 11/23 de savoir que sa donnée a été lue.

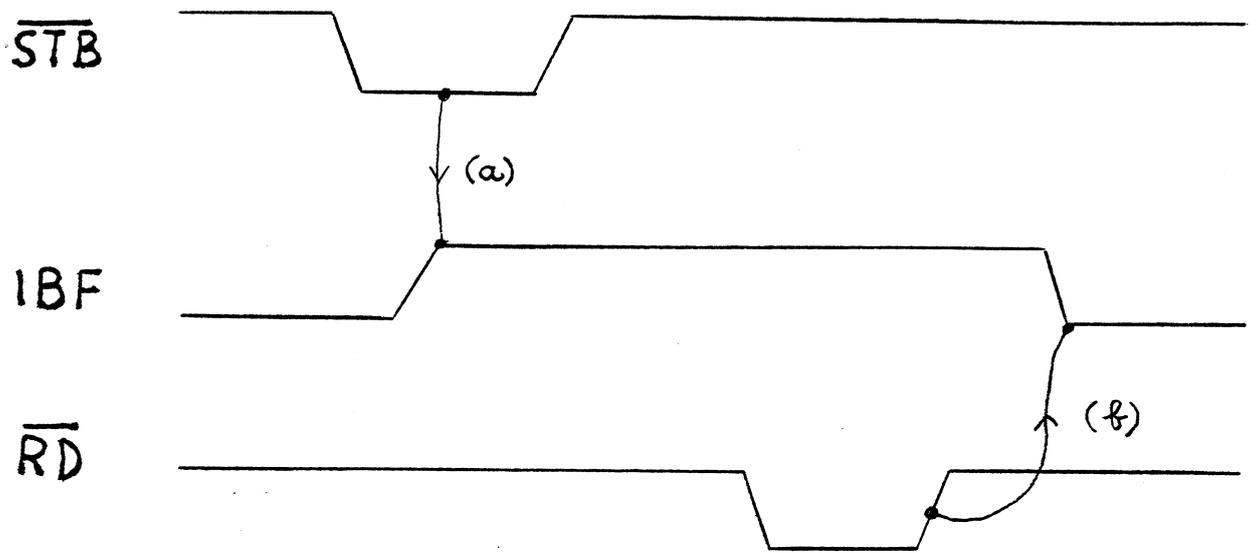


figure C.1

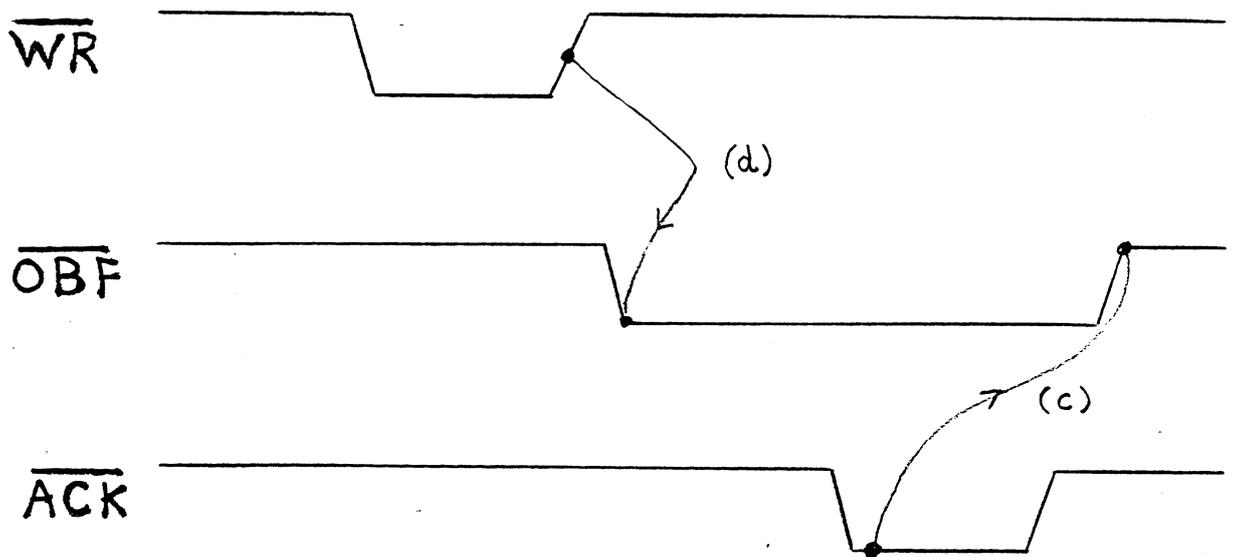


figure C.2

Aussi les programmes de transmission DRV11--> 8255 auront la forme suivante :

LSI 11/23            =====>            8080

tantque Req A = 0 attendre;  
 écrire RS;

tantque IBF = 0 attendre;  
 lire RE;

DRV11            <=====            8255

\* Data Transmitted



$\overline{\text{ACK}}$

Ce signal à impulsion (Acknowledge) est envoyé au 8255 par la carte DRV11 aussitôt que le registre d'entrée RE de la DRV11 est vidé.

\* Req B



$\overline{\text{OBF}}$

Quand le signal Data Transmitted est transmis, il met à 1 la bascule OBF (figure C.2 : c), qui permet au 8080 de savoir qu'il peut mettre une nouvelle donnée en sortie. Ce signal est renvoyé à la DRV11 (Req B) et permet au LSI 11/23 de savoir que la prochaine donnée à recevoir n'a pas encore été mise. Au moment où le 8080 écrit sur son registre de sortie, cela met à zéro la bascule OBF (figure C.2 : d). Ce signal est renvoyé à la DRV11 (Req B) et permet au LSI 11/23 de savoir que la prochaine donnée à recevoir est prête.

Aussi les programmes de transmissions 8255 ---> DRV11 auront la forme suivante :

LSI 11/23            <=====            8080

tantque Req B = 0 attendre;  
 lire RE;

tantque  $\overline{\text{OBF}}$  = 0 attendre;  
 écrire RS;

L'interface LSI 11/23 <---> 8080, du côté LSI 11/23 est réalisée à l'aide d'une seule primitive d'entrée-sortie :

I08080 (n1 ,n2)

qui réalise la fonction d'envoi de n1 octets, puis la réception de n2 octets (n1 ou n2 peuvent être nuls). Les octets transférés sont mis dans des zones mémoires globales (COMMON fortran).

## 2. Protocole d'échange

Le protocole défini pour les échanges entre le LSI 11/23 et le 8080 est de type "maître-esclave", c'est-à-dire que le séquençement des échanges d'informations entre les deux processus est absolument déterminé par le LSI 11/23 (maître). Le 8080 (esclave) est en écoute permanente du LSI 11/23 dès sa mise sous tension et travaille en mode "interruption programmée". Le micro 8080 procède à la lecture périodique du port d'entrée-sortie, afin de réagir à toute demande du calculateur maître. La transmission des informations entre les deux processus se fait par blocs d'octets. Chaque bloc est assorti de contrôles internes sur la structure des données du message.

Une information n'est prise en compte que si tous les contrôles sont effectués avec succès. Sinon, le calculateur demande la répétition du message erroné, procédure qui évite ainsi la perte des informations. Si la procédure de répétition est déclenchée 3 fois successivement, le calculateur détecte l'anomalie et prévient l'opérateur par un message de défaut.

### 3. Le logiciel implanté sur le 8080

La fonction essentielle du 8080 est de commander les déplacements du plateau translatable sur ordre du LSI 11/23. Il dispose notamment des primitives "envoi de pas", "arrêt", "changement de vitesse", "changement de sens". La boucle principale du système est représentée sur la figure C.3, et la figure C.4 montre le format d'une information reçue par le microprocesseur.

### 4. Le logiciel implanté sur le LSI 11/23

Les différentes fonctions permettant d'interagir avec les déplacements du plateau sont présentées dans l'ANNEXE B. L'ensemble est programmé de manière hiérarchique, avec appels de procédures :

* niveau 0	IO8080, TRAPAS
* niveau 1	ARRET, ENVPAS, MAJPAS
* niveau 2	AVNPAS, DEPLA, POSITN, ...

On trouve, d'une façon générale, au niveau 2, deux types de programmes qui sont les suivants :

- \* action avec attente de la fin l'action précédente.
- \* action sans attente, avec interruption programmée.

Le premier type est réalisé avec la primitive MAJPAS, qui permet une synchronisation. Cela revient à attendre que le déplacement soit terminé. On a :

```

MAJPAS
                                ---> synchronisation
ACTION

```

Le second type est réalisé avec les primitives ARRET et ENVPAS. Au moment où le programme ARRET est appelé, cela interrompt le déplacement du plateau en cours et le 8080 renvoie au LSI 11/23 le nombre de pas qui restait à faire. Une fois l'action terminée, on renvoie au 8080 le nombre de pas qui restait à faire. Ainsi il est possible de modifier la vitesse, de demander la position à tout moment, sans que le déplacement du plateau en soit altéré, on a :

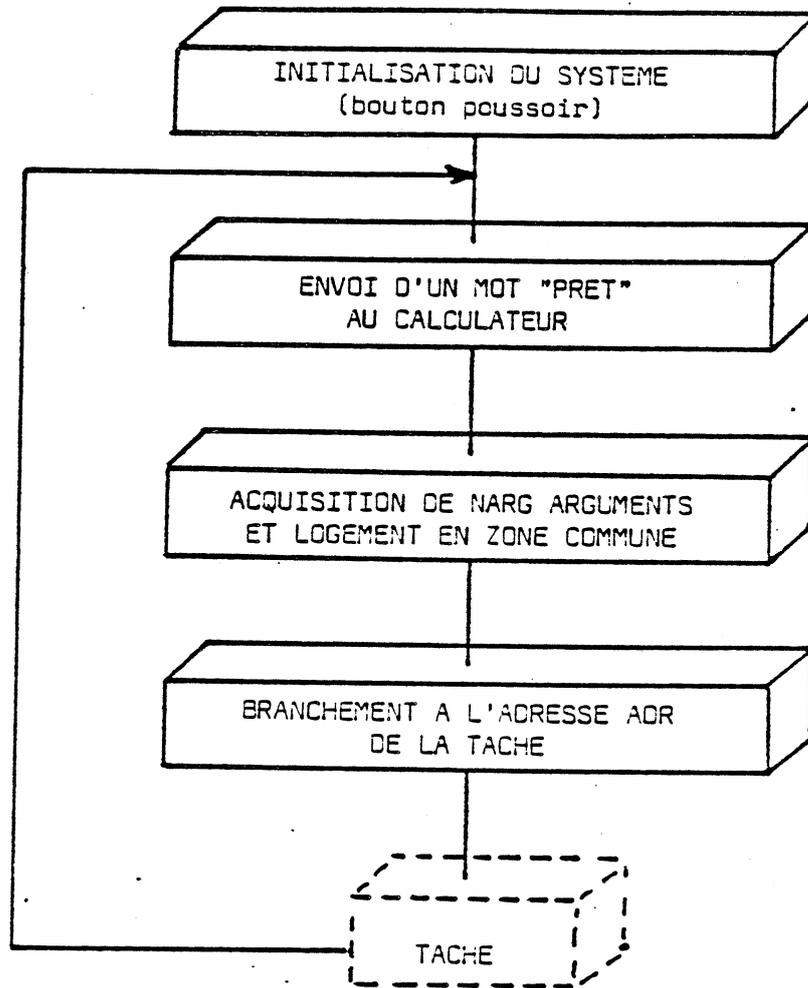


figure C.3

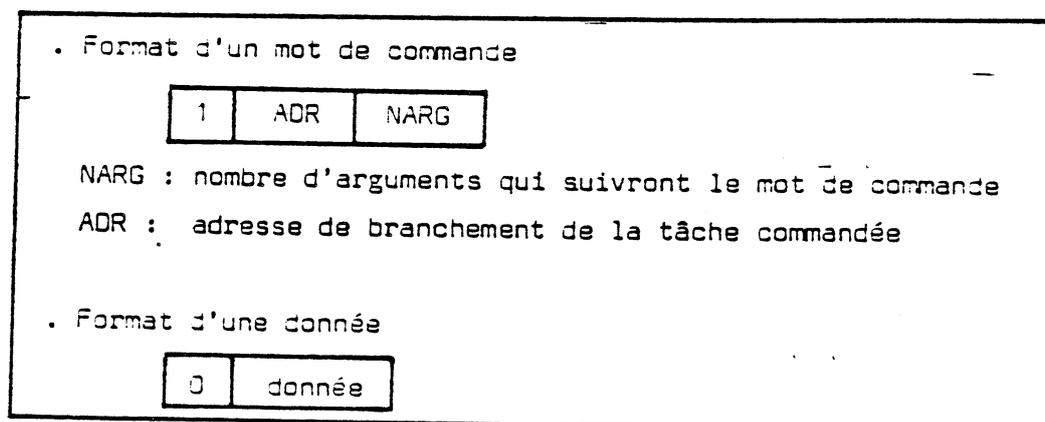


figure C.4

ARRET  
ACTION ----> interruption  
ENVPAS (le reste des pas)

## ANNEXE D

## MODELISATION MATHEMATIQUE D'UNE CAMERA

## APPLICATION AU CAPTEUR 3D

Cette annexe réunit quelques résultats relatifs à la géométrie des caméras. Certains sont bien connus et sont énoncés sans démonstration; d'autres, que nous avons mis au point, font l'objet ici d'une étude spéciale. Les deux premiers paragraphes ("spécification géométrique et fonctionnelle d'une caméra" et "calibrage d'une caméra fixe") se retrouvent dans la littérature [Duda-73] [Ballard-82]. Ils constituent la base de départ de toute description mathématique du système géométrique que constitue une caméra. Le troisième paragraphe ("relations entre spécifications fonctionnelle et géométrique") n'existe pas sous cette forme à notre connaissance, et permet de donner une base mathématique au problème du "calibrage d'une caméra mobile" qui fait l'objet du paragraphe 4. Le cinquième paragraphe est constitué d'applications des résultats présentés. La première est le cas du "calibrage plan", dont deux cas particuliers sont les capteurs 3D à plan lumineux, et la vision plane [Lorette-81]. La seconde est le cas des capteurs 3D à plan lumineux mobile par rapport à la caméra. La troisième est le cas de la stéréovision, considérée comme un cas d'utilisation simultanée de deux caméras. Le sixième paragraphe intitulé "calcul des points de fuite" donne des résultats utilisés par le chapitre 6. Le septième paragraphe regroupe des démonstrations liées aux résultats des paragraphes 3,5 et 6.

## 1. Spécification géométrique et fonctionnelle d'une caméra

Le modèle mathématique le plus simple d'une caméra est la transformation perspective donnée par la figure D.1, où le plan  $C, i, j$  est appelé plan rétinien, et  $F$  point focal.

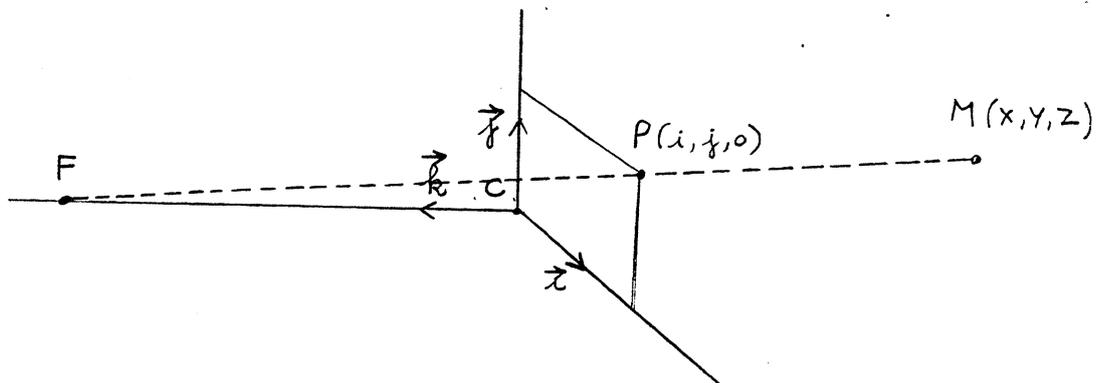


figure D.1

On trouve cette présentation dans [Ballard-82], et une présentation différente dans [Duda-73], où le point F est placé de l'autre côté du plan rétinien.

La transformation perspective, aussi appelée transformation conique, fait l'objet d'une étude dans [Carlbom-78], avec d'autres projections (orthographiques et obliques). Elle fait passer d'un point  $M:(X,Y,Z)$ , exprimé dans le repère  $(C, \vec{i}, \vec{j}, \vec{k})$  de la caméra, au point  $P:(i,j,0)$ , exprimé dans le même repère. Des considérations sur les triangles semblables permettent d'obtenir le résultat suivant:

$$(1) \quad i = \frac{f X}{f - Z} \quad j = \frac{f Y}{f - Z}$$

où  $f$  est la distance  $CF$  ( $CF=f.k$ ). Comme toute projection, cette transformation n'est évidemment pas bijective.

En coordonnées homogènes, le point  $(X,Y,Z)$  s'écrit  $(wX,wY,wZ,w)$  et le point  $(i,j,0)$  s'écrit  $(w'i,w'j,0,w')$ . La transformation perspective peut alors se représenter par la matrice  $T$  4X4 suivante:

$$(2) \quad T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/f & 1 \end{pmatrix}$$

On vérifie que les points 3D représentés par  $T*(wX,wY,wZ,w)$  et  $(w'i,w'j,0,w')$  sont équivalents. L'élément  $T(3,3)$  de cette matrice est nul: c'est normal car  $T$  n'est pas bijective. On substitue souvent ([Duda-73] et [Ballard-82]) à  $T$  la matrice  $T'$ :

$$(3) \quad T' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{pmatrix} \quad T'^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{pmatrix}$$

où l'élément  $(3,3)$  est 1.  $T'$  est inversible et son inverse est appelée "transformation perspective inverse". Cette transformation, dans son principe, permet d'associer à chaque pixel  $P:(i,j,0)$  du plan rétinien la droite de visée passant par ce pixel et le point focal F. Dans ce cas, en utilisant  $T'$ , on a:

$$(4) \quad X = \frac{f i}{f + k} \quad Y = \frac{f j}{f + k} \quad Z = \frac{f k}{f + k}$$

En faisant varier la valeur de  $k$ , on obtient l'ensemble des points de la droite de visée passant par  $P$  et  $F$ . En particulier, si  $k=0$ , on a  $M=P$  et si  $k=\infty$ ,  $M=F$ .

La modélisation ci-dessus associée à un point  $M:(X,Y,Z)$  décrit dans le repère  $(C, \vec{i}, \vec{j}, \vec{k})$  associé à la caméra un point image  $P:(i,j,0)$ , décrit aussi dans ce repère. En pratique, le point  $(X,Y,Z)$  est plutôt connu dans un autre repère  $(O, \vec{X}, \vec{Y}, \vec{Z})$  qu'on appellera repère "objet", qui se déduit du repère caméra par une transformation rigide (rotation+translation). Dans ce cas, on cherchera à exprimer la relation entre  $(X,Y,Z)$  décrit dans le repère objet et  $(i,j,0)$  décrit dans le repère caméra. Deux spécifications de caméras sont alors possibles [Prajoux-82]:

- La première (spécification géométrique) consiste à s'appuyer sur le fait que tous les rayons incidents passent par le point focal  $F$  de la caméra, et à calculer leurs intersections avec le plan rétinien. Elle consiste à donner les vecteurs  $\vec{i}$  et  $\vec{j}$  et les points  $C$  et  $F$ , en les exprimant dans le repère objet.
- La seconde (spécification fonctionnelle) fait référence à la transformation  $(X,Y,Z) \rightarrow (i,j)$ , sans faire d'hypothèse sur le trajet des rayons optiques. On montre (voir paragraphe 7.1) que la fonction qui réalise cette transformation obéit aux équations:

$$(5) \quad \begin{cases} i = \frac{a_{11}X + a_{12}Y + a_{13}Z + a_{14}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \\ j = \frac{a_{21}X + a_{22}Y + a_{23}Z + a_{24}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \end{cases}$$

Cette seconde spécification ne fait appel qu'à l'étude des fonctions. En coordonnées homogènes, le point  $(X,Y,Z)$  devient  $(wX, wY, wZ, w)$  et le pixel  $(i,j)$  devient  $(w'i, w'j, w')$ . La fonction perspective peut être alors décrite par une matrice  $A$   $3 \times 4$ :

$$(6) \quad A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

Remarque :

La modélisation de caméra ci-dessus repose sur un certain nombre d'hypothèses simplificatrices. Elle suppose d'abord que le balayage dans le tube de la caméra est linéaire (pour une caméra à matrice CCD ou CID le problème ne se pose pas). S'il ne l'est pas, il est nécessaire d'évaluer les erreurs de linéarité en  $i$  et en  $j$ , par exemple en visualisant avec la caméra un réseau de lignes régulièrement espacées dessinées sur un plan, perpendiculairement à l'axe de visée de la caméra ( $\vec{k}$ ). La modélisation suppose ensuite que les erreurs dues aux objectifs (aberrations géométriques et chromatiques) sont négligeables. Si ce n'est pas le cas, elles peuvent être corrigées par le même procédé que précédemment, en même temps que les erreurs de linéarité. Enfin, le modèle suppose que l'image d'un point est un point. Cela n'est valable que si le diaphragme de l'objectif est suffisamment fermé. Sinon, l'image d'un point  $M$  sera une tache dans le plan rétinien, la caméra se comportant comme plusieurs caméras, de points focaux légèrement

décalés. Il faut noter cependant que la diminution du diamètre du diaphragme est limitée par deux contraintes: la première est liée à la quantité de lumière nécessaire pour impressionner la rétine (sensibilité) et la seconde, négligeable, est liée aux effets de diffraction sur les bords du diaphragme.

## 2. Calibrage d'une caméra fixe

Dans le cas de la spécification géométrique, le calibrage de la caméra consiste à mesurer les vecteurs  $\vec{i}, \vec{j}$  et les points C et F, dans le repère objet. Cette méthode est plutôt difficile à mettre en oeuvre, non pas à cause des calculs (qu'on trouve dans [Duda-73]), mais parce qu'il faut obtenir expérimentalement les paramètres internes de la caméra (mise au point, zoom, etc...) qui déterminent la position de F dans le repère caméra, et les paramètres externes de la caméra (orientation et position), qui déterminent les vecteurs  $\vec{i}, \vec{j}$  et le point C dans le repère objet.

Dans le cas de la spécification fonctionnelle, le calibrage consiste à déterminer les coefficients  $a_{ij}$  de la matrice A. Pour cela, la méthode généralement utilisée consiste à résoudre un système d'équations en mettant en correspondance un ensemble de points  $M_i$  donnés dans le repère objet et leurs images  $P_i$  déterminés dans le repère caméra [Ballard-82] [Prajoux-82] [Lorette-81]. Chaque correspondance fournit deux équations (5). Le nombre d'inconnues est 11 car une des variables est libre (par exemple on peut choisir arbitrairement  $a_{34}=1$ ). Il faut donc 5 1/2 correspondances (donc 6). En pratique, on peut prendre plus de points et optimiser le résultat par les moindres carrés.

Cette opération peut se réaliser facilement en montrant à la caméra un cube dont trois des faces visibles possèdent des niveaux de gris différents (voir figure D.2).

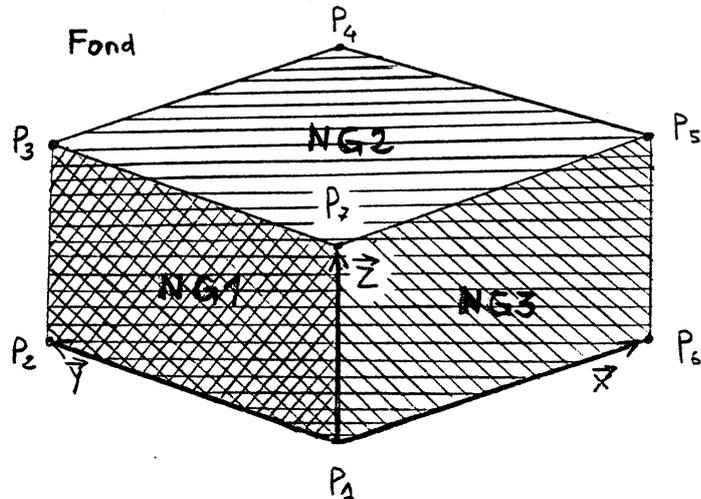


figure D.2

Les positions des points  $M_1, M_2, \dots, M_7$  (qui ont pour image les points  $P_1, P_2, \dots, P_7$  dans le repère caméra), sont, dans le repère objet, respectivement:

$$(0,0,0) (0,1,0) (0,1,1) (1,1,1) (1,0,1) (1,0,0) (0,0,1)$$

On peut repérer et identifier les noeuds  $P_i$  de deux manières. La première consiste à faire un suivi du contour extérieur de l'objet. Cela permet de repérer les points  $P_1, P_2, \dots, P_6$ , en segmentant ce contour. L'indétermination en rotation (autour de  $P_7$ ) est levée en supposant par exemple que le point  $P_1$  est le plus bas. Ces six points suffisent pour calibrer la caméra. La seconde consiste à identifier les noeuds par rapport aux niveaux de gris qui les cernent, respectivement:

$$(Fond, NG1, NG3) (Fond, NG1) (Fond, NG1, NG2) (Fond, NG2) (Fond, NG2, NG3) \\ (Fond, NG3) (NG1, NG2, NG3)$$

Une autre possibilité similaire, pour la vision en couleur, consiste à peindre chaque face avec une couleur différente.

### 3. Relation entre les spécifications géométrique et fonctionnelle d'une caméra

#### 3.1 Géométrie ---> fonctionnel

On dispose de la définition géométrique d'une caméra, et on cherche à obtenir la définition fonctionnelle. La donnée de la géométrie de la caméra comporte:

- les vecteurs  $\vec{i}$  et  $\vec{j}$  du plan rétinien, exprimés dans le repère objet. Ces vecteurs ne sont pas nécessairement normés (ce qui permet de traiter le cas des pixels rectangulaires), ni

orthogonaux. Le repère orthonormé associé à cette caméra sera donné par les vecteurs :  $\vec{i}' = \vec{i} / \|\vec{i}\|$ ,  $\vec{k}' = (\vec{i} \wedge \vec{j}) / \|\vec{i} \wedge \vec{j}\|$ ,  $\vec{j}' = \vec{k}' \wedge \vec{i}'$ .

- les points C et F, exprimés dans le repère objet.

Les coefficients  $a_{ij}$  étant ceux de la matrice A donnée par (6), on pose :

$$(7) \quad D = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \text{et} \quad E = \begin{pmatrix} a_{14} \\ a_{24} \\ a_{34} \end{pmatrix} \quad \text{et} \quad \begin{cases} \vec{P} = (a_{11} & a_{12} & a_{13}) \\ \vec{Q} = -(a_{21} & a_{22} & a_{23}) \\ \vec{R} = (a_{31} & a_{32} & a_{33}) \end{cases}$$

Les inconnues sont alors les vecteurs  $\vec{P}, \vec{Q}, \vec{R}$  (ou la matrice D), et E, on a :

$$(8) \quad \begin{cases} \vec{P} = \vec{CF} \wedge \vec{j} \\ \vec{Q} = \vec{CF} \wedge \vec{i} \\ \vec{R} = \vec{i} \wedge \vec{j} \end{cases} \quad \text{et} \quad E = -D \cdot \vec{OF}$$

La démonstration de ces résultats fait appel à des calculs donnés au paragraphe 7.1 de cette annexe.

#### Remarque:

En théorie, les vecteurs  $\vec{P}, \vec{Q}$  et  $\vec{R}$  sont connus à une homothétie près. En pratique, cela n'a pas d'importance, car cela donne la même transformation (on a déjà vu qu'une des variables de la matrice A est libre).

### 3.2 Fonctionnel $\rightarrow$ géométrique

On dispose de la définition fonctionnelle d'une caméra et on cherche à obtenir la définition géométrique. On suppose les mêmes conventions qu'au dernier paragraphe. On a :

$\vec{OF}$  est solution du système linéaire  $E = -D \cdot \vec{OF}$ , et :

$$(9) \quad \begin{cases} \vec{i} = \vec{Q} \wedge \vec{R} \\ \vec{j} = \vec{P} \wedge \vec{R} \\ \vec{CF} = \vec{Q} \wedge \vec{P} \end{cases}$$

La démonstration de ces résultats fait appel à des calculs donnés au paragraphe 7.1 de cette annexe.

#### Remarque:

Les vecteurs  $\vec{i}, \vec{j}$  et  $\vec{CF}$  sont connus à une homothétie près, en effet on a toujours la même caméra, si on passe de  $\vec{i}, \vec{j}, \vec{CF}$  à  $\theta \vec{i}, \theta \vec{j}, \theta \vec{CF}$ .

## 4. Calibrage d'une caméra mobile.

Le calibrage d'une caméra fixe a pour but l'obtention des paramètres  $a_{ij}$  de la matrice  $A$  (relation 6). Celui d'une caméra mobile consiste à obtenir les paramètres  $a_{ij}(t)$  d'une matrice  $A(t)$ .

Soient les coefficients  $a_{ij}(t)$  à un instant  $t$ . Entre  $t$  et  $t+1$ , on applique une transformation  $T_t$  à la caméra. Pour connaître les nouveaux coefficients  $a_{ij}(t+1)$ , il suffit de calculer les vecteurs  $i(t), j(t)$  et les points  $C(t)$  et  $F(t)$  de la caméra à l'instant  $t$  en utilisant les relations (9), de leur appliquer la transformation  $T_t$ , ce qui donne les vecteurs  $i(t+1), j(t+1)$  et les points  $C(t+1)$  et  $F(t+1)$ , puis, en utilisant les relations (8), de calculer les coefficients  $a_{ij}(t+1)$ . Le calibrage d'une caméra fixe (paragraphe 2) permet d'obtenir les coefficients  $a_{ij}(0)$  à l'instant  $t=0$ . Ces traitements sont résumés par la figure D.3.

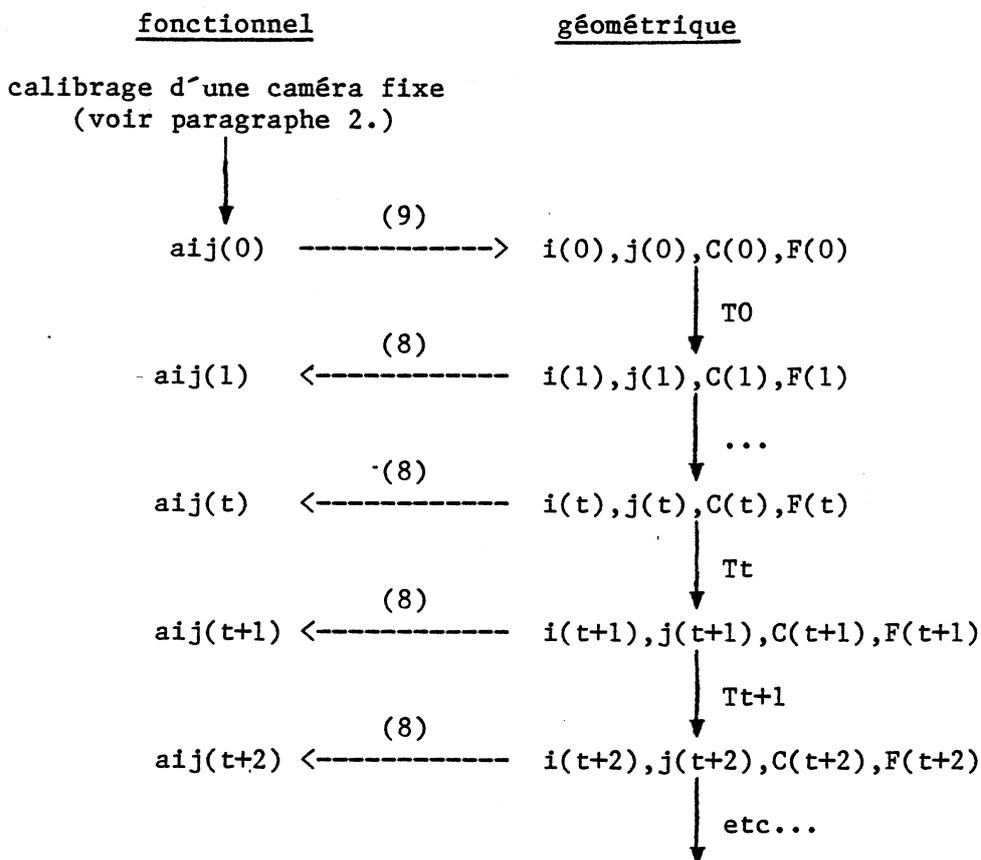


figure D.3

Remarque :

[Sobel-74] s'intéresse au cas général du calibrage d'une caméra, avec possibilité de faire varier à la fois les paramètres internes et les paramètres externes de la caméra. Mais ce cas général s'avère non totalement résolu, alors qu'en supposant les paramètres internes de la caméra constants, nous proposons une solution simple et assez aisée à mettre en oeuvre.

## 5. Applications

## 5.1 Calibrage plan

Ce que nous appellons "calibrage plan" est le procédé qui permet d'obtenir les coefficients d'une transformation qui fait passer d'un plan à un autre.

Soit un plan  $p_1$ , muni d'un repère  $(C_1, \vec{i}_1, \vec{j}_1, k_1)$  ( $p_1$  est  $k_1=0$ ), et un plan  $p_2$ , muni d'un repère  $(C_2, \vec{i}_2, \vec{j}_2, k_2)$  ( $p_2$  est  $k_2=0$ ). Soit une scène plane constituée elle aussi d'un plan  $p$ , muni d'un repère  $(O, X, Y, Z)$  ( $p$  est  $Z=0$ ) (voir figure D.4). Nous montrons que la relation entre un point  $M$  de cette scène et l'image  $P_1$  de ce point  $M$  par la caméra  $\vec{i}_1, \vec{j}_1, C_1, F_1$  est du même type que la relation entre ce point  $P_1$  et le point  $P_2$  image de  $M$  par la caméra  $\vec{i}_2, \vec{j}_2, C_2, F_2$ . Ces points ( $M$  et  $P_1$ , ou  $M$  et  $P_2$ , ou  $P_1$  et  $P_2$ ) se correspondent par homographie, qui est la transformation géométrique ponctuelle la plus générale d'un plan  $p$  vers un autre plan  $p'$  de  $\mathbb{R}^3$  [Lorette-81].

Considérons tout d'abord le cas des points  $M$  et  $P_1$ , ce qui correspond au cas de notre capteur 3D à plan lumineux: le repère de la scène est un repère attaché au plan lumineux. L'équation du plan lumineux étant  $Z=0$ , dans  $O$ , l'image d'un point du plan  $p$   $M:(X, Y, 0)$  dans  $O$  par la caméra  $C_1$  sera le point du plan  $p_1$   $P_1:(i_1, j_1, 0)$  dans  $C_1$  et on a:

$$(10) \quad \begin{cases} i_1 = \frac{a_{11}X + a_{12}Y + a_{14}}{a_{31}X + a_{32}Y + a_{34}} \\ j_1 = \frac{a_{21}X + a_{22}Y + a_{24}}{a_{31}X + a_{32}Y + a_{34}} \end{cases}$$

en utilisant la relation (5) avec  $Z=0$ . Cela définit une homographie  $H_1$  du plan  $p$  dans le plan  $p_1$ . De même, il existe une homographie  $H_2$  du plan  $p$  dans le plan  $p_2$ :

$$\begin{cases} i_2 = \frac{a'_{11}X + a'_{12}Y + a'_{14}}{a'_{31}X + a'_{32}Y + a'_{34}} \\ j_2 = \frac{a'_{21}X + a'_{22}Y + a'_{24}}{a'_{31}X + a'_{32}Y + a'_{34}} \end{cases}$$

Si l'on considère la figure D.4, on voit que la relation entre  $p$  et  $p_1$  est symétrique ainsi que la relation entre  $p$  et  $p_2$ : l'inverse d'une homographie est aussi une homographie.  $H_1$  est donc de la forme:

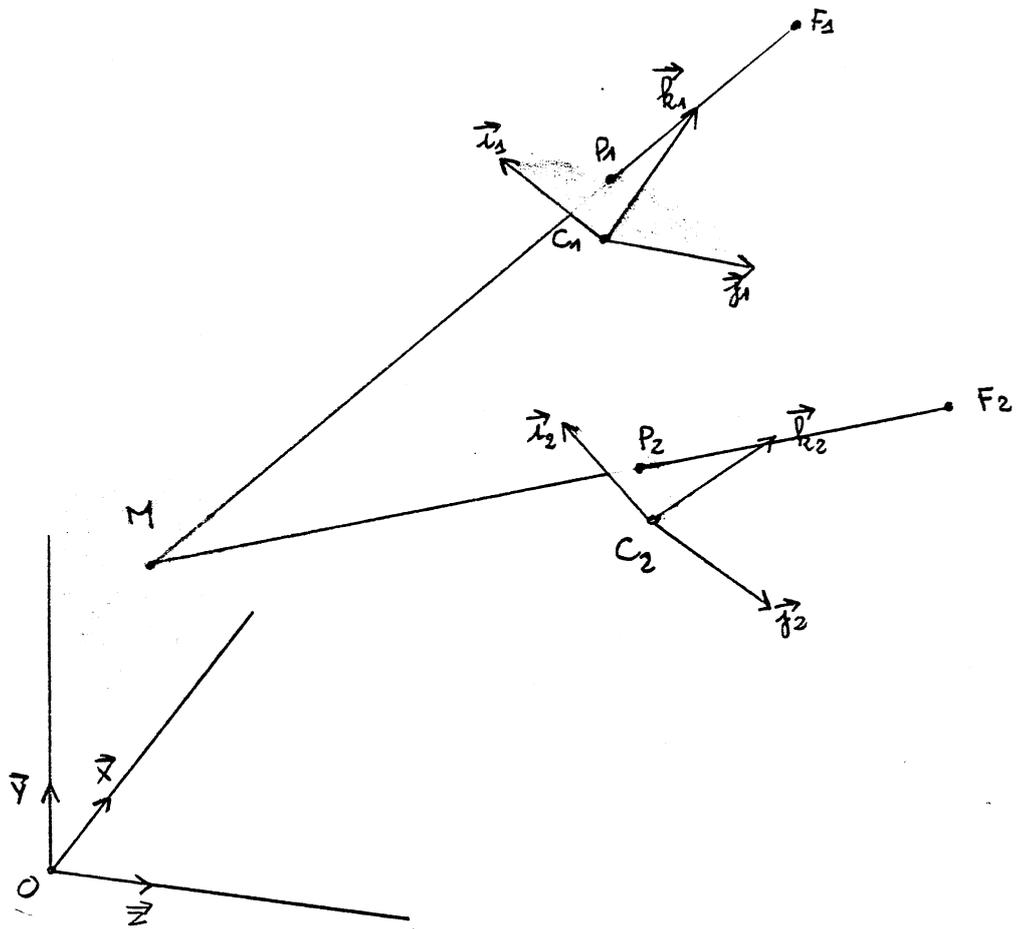


figure D.4

$$(11) \quad \begin{cases} X = \frac{b_{11}i_1 + b_{12}j_1 + b_{14}}{b_{31}i_1 + b_{32}j_1 + b_{34}} \\ Y = \frac{b_{21}i_1 + b_{22}j_1 + b_{24}}{b_{31}i_1 + b_{32}j_1 + b_{34}} \end{cases}$$

La transformation qui fait passer de  $p_1$  à  $p_2$  peut être considérée comme la composition de  $H_1$  par  $H_2$ .  $H_1$  et  $H_2$  étant des homographies, et la composition de deux homographies étant aussi une homographie, la transformation qui fait passer de  $p_1$  à  $p_2$  est donc aussi une homographie.

L'obtention des paramètres d'une homographie d'un plan  $p_1$  vers un plan  $p_2$  consiste, comme pour le paragraphe 2, à résoudre un système d'équations linéaires obtenues en mettant en correspondance un certain nombre de points de  $p_1$  avec des points de  $p_2$ . Dans ce cas, le nombre d'inconnues étant de 8 (une variable est libre), l'association de quatre couples de points permet de résoudre le problème. C'est ainsi que fonctionne le calibrage de notre capteur 3D (équations (11)). Dans le cas des capteurs stéréoscopiques, il n'existe pas de relation homographique entre les deux caméras, si elles ne visualisent pas une scène plane.

## 5.2 Capteur 3D à plan lumineux mobile

Soit  $(0, \vec{X}, \vec{Y}, \vec{Z})$  le repère objet, tel que les différents plans lumineux soient d'équations  $Z=tz$ . Les équations (5) nous donnent, pour le plan  $Z=tz$ :

$$(12) \quad \begin{cases} i = \frac{a_{11}X + a_{12}Y + a_{13}tz + a_{14}}{a_{31}X + a_{32}Y + a_{33}tz + a_{34}} \\ j = \frac{a_{21}X + a_{22}Y + a_{23}tz + a_{24}}{a_{31}X + a_{32}Y + a_{33}tz + a_{34}} \end{cases}$$

A partir de la position de l'image  $P:(i,j,0)$  du point  $M:(X,Y,tz)$  et de la valeur  $tz$ , il faut trouver  $X$  et  $Y$ . La démonstration du paragraphe 7.2 montre qu'on peut trouver des coefficients  $b_{ij}$  et  $c_{ij}$  tels que:

$$(13) \quad \begin{cases} X = \frac{(b_{11}+c_{11}tz)i + (b_{12}+c_{12}tz)j + (b_{14}+c_{14}tz)}{b_{31}i + b_{32}j + b_{34}} \\ Y = \frac{(b_{21}+c_{21}tz)i + (b_{22}+c_{22}tz)j + (b_{24}+c_{24}tz)}{b_{31}i + b_{32}j + b_{34}} \end{cases}$$

Le nombre d'inconnues est de 9 pour les coefficients  $b_{ij}$  et 6 pour les coefficients  $c_{ij}$ , ce qui donne 14 coefficients à déterminer. En effectuant deux calibrages plans tel que celui du paragraphe 5.1, pour deux valeurs différentes de Z,  $tz_1$  et  $tz_2$ , on obtient alors 8 associations, ce qui donne 16 équations. Il est donc possible de calculer les différents paramètres en résolvant un système linéaire.

### 5.3 Vision stéréoscopique

Soient deux caméras  $\vec{i}_1, \vec{j}_1, C_1, F_1$  et  $\vec{i}_2, \vec{j}_2, C_2, F_2$  inspectant une scène décrite dans un repère  $(O, \vec{X}, \vec{Y}, \vec{Z})$ . On peut écrire (voir paragraphe 1.2), si M est un point  $(X, Y, Z)$  dans O et P1 et P2 ses images par les deux caméras:

$$(14) \quad \left\{ \begin{array}{l} i_1 = \frac{a_{11}X + a_{12}Y + a_{13}Z + a_{14}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \\ j_1 = \frac{a_{21}X + a_{22}Y + a_{23}Z + a_{24}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \end{array} \right. \quad \left\{ \begin{array}{l} i_2 = \frac{a'_{11}X + a'_{12}Y + a'_{13}Z + a'_{14}}{a'_{31}X + a'_{32}Y + a'_{33}Z + a'_{34}} \\ j_2 = \frac{a'_{21}X + a'_{22}Y + a'_{23}Z + a'_{24}}{a'_{31}X + a'_{32}Y + a'_{33}Z + a'_{34}} \end{array} \right.$$

En visualisant un objet tel que celui de la figure D.2 et en calibrant séparément chacune des caméras, on obtient les paramètres  $a_{ij}$  et  $a'_{ij}$ .

La reconstitution d'un point dans l'espace M à partir de la donnée des ses images P1 et P2 par les deux caméras revient à résoudre le système de 4 équations (14) avec X, Y, Z considérées comme inconnues. On peut soit ne prendre que 3 des équations et résoudre avec un algorithme de résolution de systèmes linéaires, soit prendre les 4 équations et optimiser suivant les moindres carrés.

En général, les deux caméras sont placées de telle manière à ce que les images P1 et P2 aient le même numéro de ligne ( $j_1=j_2$ ). Dans ce cas, il suffit de prendre les 3 équations relatives à  $i_1, i_2$  et  $j_1$ , la disparité étant donnée par la valeur  $i_2-i_1$ .

## 6. Calcul des points de fuite

Toutes les droites de l'espace qui ont une certaine direction donnée  $(U, V, W)$  dans le repère objet, ont pour images des droites 2D qui se coupent toutes en un point particulier Pf, le point de fuite associé à cette direction (pour certaines directions particulières, on peut dire que le point de fuite est à l'infini).

Ce résultat, ainsi que les formules qui suivent, sont montrés au paragraphe 7.3.

Soit une direction de l'espace, donnée par le vecteur  $(U, V, W)$ , le point de fuite associé est le point Pf:  $(P_i, P_j)$ :

$$(15) \quad \begin{cases} P_i = \frac{a_{11}U + a_{12}V + a_{13}W}{a_{31}U + a_{32}V + a_{33}W} \\ P_j = \frac{a_{21}U + a_{22}V + a_{23}W}{a_{31}U + a_{32}V + a_{33}W} \end{cases}$$

où les  $a_{ij}$  sont les coefficients de la matrice A de la spécification fonctionnelle de la caméra.

Si  $P_0$  est l'image de l'origine O du repère objet, et si  $P_X, P_Y, P_Z$  sont les points de fuite associés aux directions  $\vec{X}, \vec{Y}$  et  $\vec{Z}$ , on a :

$$(16) \quad P_X = \left( \frac{a_{11} \ a_{21}}{a_{31} \ a_{31}}, \frac{a_{12} \ a_{22}}{a_{32} \ a_{32}} \right) \quad P_Y = \left( \frac{a_{12} \ a_{22}}{a_{32} \ a_{32}}, \frac{a_{13} \ a_{23}}{a_{33} \ a_{33}} \right) \quad P_Z = \left( \frac{a_{13} \ a_{23}}{a_{33} \ a_{33}}, \frac{a_{14} \ a_{24}}{a_{34} \ a_{34}} \right) \quad P_0 = \left( \frac{a_{14} \ a_{24}}{a_{34} \ a_{34}}, \frac{a_{15} \ a_{25}}{a_{35} \ a_{35}} \right)$$

Ces résultats sont particulièrement intéressants si l'on souhaite visualiser, en superposition sur l'image fournie par la caméra, des éléments graphiques tels que points et droites. En particulier, cela permet de représenter l'image par la caméra du repère objet.

Inversement, on peut associer à tout point du plan  $C, \vec{i}, \vec{j}$  une direction de l'espace. Si on pose :

$$(17) \quad \begin{cases} \vec{P}_f = (P_i, P_j, 1) \\ \vec{C}_1 = (a_{11}, a_{21}, a_{31}) \\ \vec{C}_2 = (a_{12}, a_{22}, a_{32}) \\ \vec{C}_3 = (a_{13}, a_{23}, a_{33}) \end{cases} \quad (\vec{C}_i \text{ ne pas confondre avec les vecteurs } \vec{P}, \vec{Q} \text{ et } \vec{R} \text{ de (7)})$$

On a :

$$(18) \quad \begin{cases} \vec{U} = (\vec{C}_3 \wedge \vec{C}_2) \cdot \vec{P}_f \\ \vec{V} = (\vec{C}_1 \wedge \vec{C}_3) \cdot \vec{P}_f \\ \vec{W} = (\vec{C}_2 \wedge \vec{C}_1) \cdot \vec{P}_f \end{cases}$$

## 7. Démonstrations

### 7.1 Démonstration 1

Nous faisons ici les calculs qui permettent d'établir les formules données aux paragraphes 3.1 et 3.2.

On part de la définition géométrique d'une caméra, qui est la donnée de deux vecteurs  $\vec{i}$  et  $\vec{j}$  et de deux points C et F, exprimés

dans le repère objet  $(0, \vec{X}, \vec{Y}, \vec{Z})$ . M est un point  $(X, Y, Z)$  exprimé dans ce repère, et  $P:(i, j, 0)$  est son image dans le repère caméra.

On pose  $\vec{V} = \vec{CM}$  et  $\vec{F} = \vec{CF}$ . M, P et F sont alignés, ce qui est équivalent à dire que  $\vec{MP}$  et  $\vec{MF}$  sont colinéaires, ou encore que  $\vec{MF} \wedge \vec{MP} = \vec{0}$ .

Ce qui donne, car  $\vec{MF} = \vec{CF} - \vec{CM}$  et  $\vec{MP} = \vec{CP} - \vec{CM}$ :

$$\begin{aligned} & (\vec{CF} - \vec{CM}) \wedge (\vec{CP} - \vec{CM}) = \vec{0} \\ \text{d'où, avec } \vec{V} \text{ et } \vec{F}: & (\vec{F} - \vec{V}) \wedge (\vec{CP} - \vec{V}) = \vec{0} \end{aligned}$$

$$\text{d'où} \quad (\vec{F} - \vec{V}) \wedge \vec{CP} = (\vec{F} - \vec{V}) \wedge \vec{V} = \vec{F} \wedge \vec{V}$$

Or le point P a pour coordonnées  $(i, j, 0)$ , ce qui se traduit par  $\vec{CP} = i \cdot \vec{i} + j \cdot \vec{j}$ . On obtient donc:

$$(19) \quad (\vec{F} - \vec{V}) \wedge (i \cdot \vec{i} + j \cdot \vec{j}) = \vec{F} \wedge \vec{V}$$

On veut calculer i. C'est possible, en utilisant la relation:

$$(20) \quad (\vec{A} \wedge \vec{B}) \cdot \vec{C} = \vec{A} \cdot (\vec{B} \wedge \vec{C})$$

qui est connue. En multipliant les deux termes de (19) par  $\vec{j}$ , on obtient:

$$\begin{aligned} [(\vec{F} - \vec{V}) \wedge (i \cdot \vec{i} + j \cdot \vec{j})] \cdot \vec{j} &= (\vec{F} \wedge \vec{V}) \cdot \vec{j} = -(\vec{V} \wedge \vec{F}) \cdot \vec{j} \\ (\vec{F} - \vec{V}) \cdot [(i \cdot \vec{i} + j \cdot \vec{j}) \wedge \vec{j}] &= -\vec{V} \cdot (\vec{F} \wedge \vec{j}) \\ (\vec{V} - \vec{F}) \cdot (i \cdot \vec{i} \wedge \vec{j}) &= \vec{V} \cdot (\vec{F} \wedge \vec{j}) \end{aligned}$$

donc

$$(21) \quad i = \frac{(\vec{F} \wedge \vec{j}) \cdot \vec{V}}{(\vec{i} \wedge \vec{j}) \cdot (\vec{V} - \vec{F})}$$

Pour calculer j, on multiplie les deux membres de (19) par  $\vec{i}$ :

$$\begin{aligned} [(\vec{F} - \vec{V}) \wedge (i \cdot \vec{i} + j \cdot \vec{j})] \cdot \vec{i} &= (\vec{F} \wedge \vec{V}) \cdot \vec{i} = -(\vec{V} \wedge \vec{F}) \cdot \vec{i} \\ (\vec{F} - \vec{V}) \cdot [(i \cdot \vec{i} + j \cdot \vec{j}) \wedge \vec{i}] &= -\vec{V} \cdot (\vec{F} \wedge \vec{i}) \\ (\vec{V} - \vec{F}) \cdot (j \cdot \vec{j} \wedge \vec{i}) &= \vec{V} \cdot (\vec{F} \wedge \vec{i}) \end{aligned}$$

donc

$$(22) \quad j = \frac{(\vec{F} \wedge \vec{i}) \cdot \vec{V}}{(\vec{j} \wedge \vec{i}) \cdot (\vec{V} - \vec{F})}$$

Posons:

$$(8) \quad \begin{cases} \vec{P} = \vec{F} \wedge \vec{j} \\ \vec{Q} = \vec{F} \wedge \vec{i} \\ \vec{R} = \vec{i} \wedge \vec{j} \end{cases}$$

On obtient, avec (21) et (22):

$$(23) \quad i = \frac{\vec{P} \cdot \vec{V}}{\vec{R} \cdot (\vec{V} - \vec{F})} \quad j = - \frac{\vec{Q} \cdot \vec{V}}{\vec{R} \cdot (\vec{V} - \vec{F})}$$

On peut trouver  $\vec{i}, \vec{j}$  et  $F$ , en fonction de  $\vec{P}, \vec{Q}$  et  $\vec{R}$ , en utilisant la relation (démontrée au paragraphe 7.4):

$$(24) \quad (A \wedge B) \wedge (A \wedge C) = [A \cdot (B \wedge C)] \cdot A$$

- avec  $\vec{i}, \vec{j}$  et  $\vec{F}$ , on a:

$$\begin{aligned} (\vec{i} \wedge \vec{j}) \wedge (\vec{i} \wedge \vec{F}) &= [\vec{i} \cdot (\vec{j} \wedge \vec{F})] \cdot \vec{i} \\ \vec{R} \wedge (-\vec{Q}) &= [(\vec{i} \wedge \vec{j}) \cdot \vec{F}] \cdot \vec{i} \\ -\vec{R} \wedge \vec{Q} &= (\vec{R} \cdot \vec{F}) \cdot \vec{i} \end{aligned}$$

ce qui donne:

$$(25) \quad \vec{i} = \frac{\vec{Q} \wedge \vec{R}}{\vec{F} \cdot \vec{R}}$$

- avec  $\vec{j}, \vec{i}$  et  $\vec{F}$ , on a:

$$\begin{aligned} (\vec{j} \wedge \vec{i}) \wedge (\vec{j} \wedge \vec{F}) &= [\vec{j} \cdot (\vec{i} \wedge \vec{F})] \cdot \vec{j} \\ -\vec{R} \wedge (-\vec{P}) &= [(\vec{j} \wedge \vec{i}) \cdot \vec{F}] \cdot \vec{j} \\ \vec{R} \wedge \vec{P} &= (-\vec{R} \cdot \vec{F}) \cdot \vec{j} \end{aligned}$$

ce qui donne:

$$(26) \quad \vec{j} = \frac{\vec{P} \wedge \vec{R}}{\vec{F} \cdot \vec{R}}$$

- avec  $\vec{F}, \vec{i}$  et  $\vec{j}$ , on a:

$$\begin{aligned} (\vec{F} \wedge \vec{i}) \wedge (\vec{F} \wedge \vec{j}) &= [\vec{F} \cdot (\vec{i} \wedge \vec{j})] \cdot \vec{F} \\ \vec{Q} \wedge \vec{P} &= (\vec{F} \cdot \vec{R}) \cdot \vec{F} \end{aligned}$$

ce qui donne:

$$(27) \quad \vec{F} = \frac{\vec{Q} \wedge \vec{P}}{\vec{F} \cdot \vec{R}}$$

Il ne reste plus qu'à calculer le produit  $\vec{F} \cdot \vec{R}$ . Avec (27), en multipliant par  $\vec{R}$ , on a:

$$\vec{F} \cdot \vec{R} = \frac{\vec{Q} \wedge \vec{P}}{\vec{F} \cdot \vec{R}} \cdot \vec{R} = \frac{(\vec{Q} \wedge \vec{P}) \cdot \vec{R}}{\vec{F} \cdot \vec{R}}$$

donc

$$(\vec{F} \cdot \vec{R})^2 = (\vec{Q} \wedge \vec{P}) \cdot \vec{R}$$

En posant  $f = \vec{F} \cdot \vec{R}$  comme variable intermédiaire, on a:

$$(28) \quad \begin{cases} \vec{P} = \vec{F} \wedge \vec{j} \\ \vec{Q} = \vec{F} \wedge \vec{i} \\ \vec{R} = \vec{i} \wedge \vec{j} \\ f = \vec{F} \cdot \vec{R} \end{cases} \quad \begin{cases} f^2 = (\vec{Q} \wedge \vec{P}) \cdot \vec{R} \\ \vec{i} = (\vec{Q} \wedge \vec{R}) / f \\ \vec{j} = (\vec{P} \wedge \vec{R}) / f \\ \vec{F} = (\vec{Q} \wedge \vec{P}) / f \end{cases}$$

Les deux solutions qu'on trouve pour la valeur  $f$  correspondent à deux cas, l'un où l'image est "à l'endroit", l'autre où l'image

est à "l'envers", le point focal étant d'un côté du plan rétinien, ou de l'autre.

Les coordonnées de P, i et j, dépendent des paramètres  $\vec{P}, \vec{Q}$  et  $\vec{R}$  à un rapport  $\theta$  près. Si on remplace respectivement  $\vec{P}, \vec{Q}$  et  $\vec{R}$  par  $\theta\vec{P}, \theta\vec{Q}$  et  $\theta\vec{R}$ , i et j gardent la même valeur (23). Par contre, les vecteurs  $\vec{i}, \vec{j}$  et  $\vec{F}$  calculés avec (28) changent si on remplace  $\vec{P}, \vec{Q}$  et  $\vec{R}$  par  $\theta\vec{P}, \theta\vec{Q}$  et  $\theta\vec{R}$ . Mais le point F ne varie pas, comme le montrent les calculs qui suivent. Montrons d'abord le résultat suivant:

$$(29) \quad i = \frac{\vec{P} \cdot \vec{F} \vec{M}}{\vec{R} \cdot \vec{F} \vec{M}} \quad j = - \frac{\vec{Q} \cdot \vec{F} \vec{M}}{\vec{R} \cdot \vec{F} \vec{M}}$$

En effet, on a (23) et:

$$- \vec{F} = \vec{C} \vec{F}, \quad \vec{V} = \vec{C} \vec{M} \text{ donc } \vec{V} - \vec{F} = \vec{C} \vec{M} - \vec{C} \vec{F} = \vec{F} \vec{M}$$

$$- \vec{Q} \cdot \vec{V} = \vec{Q} \cdot \vec{C} \vec{M} = \vec{Q} \cdot (\vec{C} \vec{F} + \vec{C} \vec{M}) = \vec{Q} \cdot \vec{F} + \vec{Q} \cdot \vec{F} \vec{M}$$

or  $\vec{Q} = \vec{F} \wedge \vec{i}$  donc  $\vec{Q} \cdot \vec{F} = 0$  donc

$$\vec{Q} \cdot \vec{V} = \vec{Q} \cdot \vec{F} \vec{M}$$

$$- \vec{P} \cdot \vec{V} = \vec{P} \cdot \vec{C} \vec{M} = \vec{P} \cdot (\vec{C} \vec{F} + \vec{C} \vec{M}) = \vec{P} \cdot \vec{F} + \vec{P} \cdot \vec{F} \vec{M}$$

or  $\vec{P} = \vec{F} \wedge \vec{j}$  donc  $\vec{P} \cdot \vec{F} = 0$  donc

$$\vec{P} \cdot \vec{V} = \vec{P} \cdot \vec{F} \vec{M}$$

Maintenant que (29) est montré, faisons apparaître le vecteur  $\vec{O} \vec{F}$ . On a  $\vec{F} \vec{M} = \vec{O} \vec{M} - \vec{O} \vec{F}$ , donc:

$$(30) \quad i = \frac{\vec{P} \cdot \vec{O} \vec{M} - \vec{P} \cdot \vec{O} \vec{F}}{\vec{R} \cdot \vec{O} \vec{M} - \vec{R} \cdot \vec{O} \vec{F}} = \frac{\vec{P} \cdot \vec{O} \vec{M} + s}{\vec{R} \cdot \vec{O} \vec{M} + u}, \quad j = - \frac{\vec{Q} \cdot \vec{O} \vec{M} - \vec{Q} \cdot \vec{O} \vec{F}}{\vec{R} \cdot \vec{O} \vec{M} - \vec{R} \cdot \vec{O} \vec{F}} = - \frac{\vec{Q} \cdot \vec{O} \vec{M} + t}{\vec{R} \cdot \vec{O} \vec{M} + u}$$

$\vec{O} \vec{F}$  est solution du système de 3 équations 3 inconnues suivant:

$$(31) \quad \begin{cases} \vec{P} \cdot \vec{O} \vec{F} = -s \\ \vec{Q} \cdot \vec{O} \vec{F} = -t \\ \vec{R} \cdot \vec{O} \vec{F} = -u \end{cases}$$

Si on passe de  $\vec{P}, \vec{Q}, \vec{R}, s, t, u$  à  $\theta\vec{P}, \theta\vec{Q}, \theta\vec{R}, \theta s, \theta t$  et  $\theta u$ , on obtient un système équivalent. Le système (31) n'a qu'une solution, sinon  $\vec{P}, \vec{Q}$  et  $\vec{R}$  seraient linéairement dépendants, donc on aurait  $\vec{R} \cdot (\vec{P} \wedge \vec{Q}) = 0$ , ce qui donne:

$$\vec{R} \cdot [(\vec{F} \wedge \vec{j}) \wedge (\vec{F} \wedge \vec{i})] = 0$$

$$\vec{R} \cdot [(\vec{F} \cdot (\vec{j} \wedge \vec{i})) \vec{F}] = 0 \quad (\text{en utilisant (24)})$$

$$\vec{R} \cdot [(\vec{F} \cdot (-\vec{R})) \vec{F}] = \vec{R} \cdot [(-\vec{F} \cdot \vec{R}) \vec{F}] = -\vec{F} \cdot \vec{R} = 0$$

$$\vec{F} \cdot (\vec{i} \wedge \vec{j}) = 0$$

Cela signifie que F est dans le plan rétinien, ce qui n'est pas possible pour une caméra.

Voyons maintenant comment on prouve les résultats (8) et (9). Si on regarde la formule (30), on voit que les vecteurs  $\vec{P}, \vec{Q}$  et  $\vec{R}$  correspondent bien à la définition (7). Par conséquent la relation (8) qui était considérée comme hypothèse est juste. (9) se montre en

considérant (28) et en imposant  $f=1$ , ce qui est possible car les vecteurs  $\vec{i}$  et  $\vec{j}$  sont connus à une homothétie près. Il ne reste plus qu'à montrer la relation  $E=-D.OF$  (8). On a:  $a_{14}=s=-\vec{P}.OF$ ,  $a_{24}=-t=\vec{Q}.OF$ ,  $a_{34}=u=-\vec{R}.OF$  par la relation (30). Les lignes de la matrice  $D$  étant les vecteurs  $\vec{P}$ ,  $-\vec{Q}$  et  $\vec{R}$ , le vecteur  $(a_{14}, a_{24}, a_{34})$  étant égal à  $E$ , cela se réécrit  $E=-D.OF$ .

## 7.2 Démonstration 2

Il s'agit ici de démontrer le résultat (13) relatif aux capteurs à plan lumineux mobile. Pour cela, il faut résoudre le système (12), où les données sont  $i, j, tz$  et les inconnues  $X$  et  $Y$ . En multipliant par les dénominateurs, on obtient:

$$\begin{aligned} i(a_{31}X+a_{32}Y+a_{33}tz+a_{34}) &= a_{11}X+a_{12}Y+a_{13}tz+a_{14} \\ j(a_{31}X+a_{32}Y+a_{33}tz+a_{34}) &= a_{21}X+a_{22}Y+a_{23}tz+a_{24} \end{aligned}$$

Cela donne, en développant:

$$\begin{aligned} (a) \quad X(ia_{31}-a_{11})+Y(ia_{32}-a_{12}) &= tz(a_{13}-ia_{33})+(a_{14}-ia_{34}) \\ (b) \quad X(ja_{31}-a_{21})+Y(ja_{32}-a_{22}) &= tz(a_{23}-ja_{33})+(a_{24}-ja_{34}) \end{aligned}$$

En soustrayant à (a)\*( $ja_{32}-a_{22}$ ), (b)\*( $ia_{32}-a_{12}$ ) on obtient:

$$\begin{aligned} X[(ia_{31}-a_{11})(ja_{32}-a_{22})-(ja_{31}-a_{21})(ia_{32}-a_{12})] &= \\ [tz(a_{13}-ia_{33})+(a_{14}-ia_{34})](ja_{32}-a_{22})- & \\ [tz(a_{23}-ja_{33})+(a_{24}-ja_{34})](ia_{32}-a_{12}) &= D X = N \end{aligned}$$

Le terme  $D$  correspond au déterminant du système donné par les équations (a) et (b). Ce terme va être le dénominateur  $b_{31}i + b_{32}j + b_{34}$  de la formule (13). Pour que cela soit possible, il suffit de montrer qu'il n'y a pas de terme en  $ij$ . Si on développe l'expression  $D$  et qu'on ne garde que les termes en  $ij$ , on trouve:

$$ia_{31} ja_{32} - ja_{31} ia_{32}, \text{ ce qui est bien nul.}$$

Développons maintenant le terme  $N$ . On obtient:

$$\begin{aligned} tza_{13}ja_{32}-tza_{13}ja_{32}+a_{14}ja_{32}-ia_{34}ja_{32}+ & \\ -tza_{13}a_{22}+tza_{13}a_{22}-a_{14}a_{22}+ia_{34}a_{22} - & \\ [tza_{23}ia_{32}-tza_{23}ia_{32}+a_{24}ia_{32}-ja_{34}ia_{32}+ & \\ -tza_{23}a_{12}+tza_{23}a_{12}-a_{24}a_{12}+ja_{34}a_{12}] = & \end{aligned}$$

$$\begin{aligned} i (tza_{33}a_{22}+a_{34}a_{22}-tza_{23}a_{32}-a_{24}a_{32})+ & \\ j (tza_{13}a_{32}+a_{14}a_{32}-tza_{33}a_{12}-a_{34}a_{12})+ & \\ ij(-tza_{33}a_{32}-a_{34}a_{32}+tza_{33}a_{32}+a_{34}a_{32})+ & \\ (-tza_{13}a_{22}-a_{14}a_{22}+tza_{23}a_{12}+a_{24}a_{12}) & = \end{aligned}$$

$$\begin{aligned} [ (a_{34}a_{22}-a_{24}a_{32})+(a_{33}a_{22}-a_{23}a_{32})tz ] i + & \\ [ (a_{14}a_{32}-a_{34}a_{12})+(a_{13}a_{32}-a_{33}a_{12})tz ] j + & \\ (a_{24}a_{12}-a_{14}a_{22})+(a_{23}a_{12}-a_{13}a_{22})tz & \end{aligned}$$

Par conséquent la forme (13) est valable pour  $X$ . Par symétrie, il en est de même avec  $Y$ .

## 7.3 Démonstration 3

Il s'agit de montrer ici les formules (15), (16) et (18) du paragraphe 6.2. Montrons que toutes les droites de l'espace qui ont la direction  $(U, V, W)$  ont leurs images qui se coupent en un point  $P_f: (P_i, P_j)$  vérifiant (15). La démonstration est analytique. On considère comme un résultat connu le fait que l'image d'une droite 3D par une caméra est une droite 2D.

Soient deux points de l'espace  $M$  et  $N$  et un vecteur  $\vec{u} = (A, B, C)$ . On pose  $M' = M + \vec{u}$  et  $N' = N + \vec{u}$ . La droite  $D$  passant par  $M$  et  $M'$  est parallèle à la droite  $D'$  passant par  $N$  et  $N'$ . Le but de la démonstration est de calculer les équations des droites  $E$  et  $E'$  images de  $D$  et  $D'$  par la caméra, et de montrer qu'elles se coupent en un point qui ne dépend pas des points  $M$  et  $N$  et donc seulement de l'orientation du vecteur  $\vec{u}$ .

Si on appelle  $P, Q, P', Q'$  les images des points  $M, N, M', N'$  par la caméra, on a :

$$P: \begin{cases} i_p = \frac{a_{11}X_m + a_{12}Y_m + a_{13}Z_m + a_{14}}{a_{31}X_m + a_{32}Y_m + a_{33}Z_m + a_{34}} \\ j_p = \frac{a_{21}X_m + a_{22}Y_m + a_{23}Z_m + a_{24}}{a_{31}X_m + a_{32}Y_m + a_{33}Z_m + a_{34}} \end{cases}$$

$$Q: \begin{cases} i_q = \frac{a_{11}X_n + a_{12}Y_n + a_{13}Z_n + a_{14}}{a_{31}X_n + a_{32}Y_n + a_{33}Z_n + a_{34}} \\ j_q = \frac{a_{21}X_n + a_{22}Y_n + a_{23}Z_n + a_{24}}{a_{31}X_n + a_{32}Y_n + a_{33}Z_n + a_{34}} \end{cases}$$

$$P': \begin{cases} i_{p'} = \frac{a_{11}X_{m'} + a_{12}Y_{m'} + a_{13}Z_{m'} + a_{14}}{a_{31}X_{m'} + a_{32}Y_{m'} + a_{33}Z_{m'} + a_{34}} \\ j_{p'} = \frac{a_{21}X_{m'} + a_{22}Y_{m'} + a_{23}Z_{m'} + a_{24}}{a_{31}X_{m'} + a_{32}Y_{m'} + a_{33}Z_{m'} + a_{34}} \end{cases}$$

$$Q': \begin{cases} i_{q'} = \frac{a_{11}X_{n'} + a_{12}Y_{n'} + a_{13}Z_{n'} + a_{14}}{a_{31}X_{n'} + a_{32}Y_{n'} + a_{33}Z_{n'} + a_{34}} \\ j_{q'} = \frac{a_{21}X_{n'} + a_{22}Y_{n'} + a_{23}Z_{n'} + a_{24}}{a_{31}X_{n'} + a_{32}Y_{n'} + a_{33}Z_{n'} + a_{34}} \end{cases}$$

Pour simplifier les calculs, on posera:

$$\begin{cases} Am=a_{11}X_m+a_{12}Y_m+a_{13}Z_m+a_{14} \\ Em=a_{21}X_m+a_{22}Y_m+a_{23}Z_m+a_{24} \\ Cm=a_{31}X_m+a_{32}Y_m+a_{33}Z_m+a_{34} \end{cases}$$

$$\begin{cases} An=a_{11}X_n+a_{12}Y_n+a_{13}Z_n+a_{14} \\ En=a_{21}X_n+a_{22}Y_n+a_{23}Z_n+a_{24} \\ Cn=a_{31}X_n+a_{32}Y_n+a_{33}Z_n+a_{34} \end{cases}$$

$$\begin{cases} Au=a_{11}U+a_{12}V+a_{13}W \\ Eu=a_{21}U+a_{22}V+a_{23}W \\ Cu=a_{31}U+a_{32}V+a_{33}W \end{cases}$$

On a  $X_m' = X_m + U$ ,  $Y_m' = Y_m + V$ ,  $Z_m' = Z_m + W$ , et  
 $X_n' = X_n + U$ ,  $Y_n' = Y_n + V$ ,  $Z_n' = Z_n + W$ .

Le calcul des valeurs  $im, jm, in, jn, im', jm', in', jn'$  donne alors:

$$\begin{aligned} ip &= Am/Cm & iq &= An/Cn & ip' &= (Am+Au)/(Cm+Cu) & iq' &= (An+Au)/(Cn+Cu) \\ jp &= Em/Cm & jq &= En/Cn & jp' &= (Em+Eu)/(Cm+Cu) & jq' &= (En+Eu)/(Cn+Cu) \end{aligned}$$

Maintenant, il faut calculer l'intersection de la droite passant par P et P' et de la droite passant par Q et Q'. Pour cela il faut calculer les équations de ces deux droites.

- Droite P P', on a:  $(P_i - ip)(j_p' - j_p) = (P_j - j_p)(i_p' - ip)$ , ce qui donne:

$$P_i(j_p' - j_p) - P_j(i_p' - ip) = ip(j_p' - j_p) - j_p(i_p' - ip) = ipj_p' - jpip'$$

- Droite Q Q':  $P_i(j_q' - j_q) - P_j(i_q' - iq) = iqj_q' - jqiq'$

On veut calculer leur intersection. Pour cela, il faut résoudre le système linéaire constitué des deux équations de droite. Procédons par élimination.

$$P_i(j_p' - j_p)(i_q' - iq) - i(j_q' - j_q)(i_p' - ip) = (ipj_p' - jpip')(i_q' - iq) - (iqj_q' - jqiq')(i_p' - ip)$$

$$P_i = N/D$$

$$N = (ipj_p' - jpip')(i_q' - iq) - (iqj_q' - jqiq')(i_p' - ip)$$

$$D = (j_p' - j_p)(i_q' - iq) - (j_q' - j_q)(i_p' - ip)$$

-Effectuons le calcul de  $(ipj_p' - jpip')$

$$(ipj_p' - jpip') = \frac{Am (Em+Eu)}{Cm (Cm+Cu)} - \frac{Em (Am+Au)}{Cm (Cm+Cu)} = \frac{AmEu - EmAu}{Cm(Cm+Cu)}$$

-Calcul de  $(i_q' - iq)$

$$(i_q' - iq) = \frac{(An+Au)}{(Cn+Cu)} - \frac{An (An+Au)Cn - (Cn+Cu)An}{Cn (Cn+Cu)Cn} = \frac{AuCn - CuAn}{(Cn+Cu)Cn}$$

-Calcul de  $(ipj_p' - jpip')(i_q' - iq)$

$$(ipjp^{\sim}-jpip^{\sim})(iq^{\sim}-iq) = \frac{(AmEu-EmAu)(AuCn-CuAn)}{Cm(Cm+Cu)(Cn+Cu)Cn}$$

Si on pose  $Cm(Cm+Cu)Cn(Cn+Cu)=d$ , on a

$$(ipjp^{\sim}-jpip^{\sim})(iq^{\sim}-iq) = (AmEu-EmAu)(AuCn-CuAn)/d$$

De même, on a :

$$(iqjq^{\sim}-jqjq^{\sim})(ip^{\sim}-ip) = (AnEu-EnAu)(AuCm-CuAm)/d$$

(en inversant p et q, m et n)

-Calcul de N:

$$\begin{aligned} N &= [(AmEu-EmAu)(AuCn-CuAn) - (AnEu-EnAu)(AuCm-CuAm)]/d \\ &= [AmEuAuCn - AmEuCuAn - EmAuAuCn + EmAuCuAn \\ &\quad - (AnEuAuCm - AnEuCuAm - EnAuAuCm + EnAuCuAm)]/d \\ &= [AmEuCn - EmAuCn + EmCuAn - AnEuCm + EnAuCm - EnCuAm]Au/d \\ &= [Au(EnCm - EmCn) + Eu(AmCn - AnCm) + Cu(EmAn - EnAm)]Au/d \\ &= K Au/d \end{aligned}$$

Maintenant, calculons le dénominateur D.

$$D = (jp^{\sim} - jp)(iq^{\sim} - iq) - (jq^{\sim} - jq)(ip^{\sim} - ip)$$

-Calcul de  $(jp^{\sim} - jp)$

$$(jp^{\sim} - jp) = \frac{(Em+Eu)}{(Cm+Cu)} - \frac{Em}{Cm} = \frac{Em(Em+Eu)Cm - (Cm+Cu)Em}{(Cm+Cu)Cm} = \frac{EuCm - CuEm}{(Cm+Cu)Cm}$$

-Calcul de  $(jp^{\sim} - jp)(iq^{\sim} - iq)$

$$(jp^{\sim} - jp)(iq^{\sim} - iq) = \frac{EuCm - CuEm}{(Cm+Cu)Cm} \frac{AuCn - CuAn}{(Cn+Cu)Cn} = (EuCm - CuEm)(AuCn - CuAn)/d$$

-Calcul de  $(jq^{\sim} - jq)(ip^{\sim} - ip)$

$$(jq^{\sim} - jq)(ip^{\sim} - ip) = (EuCn - CuEn)(AuCm - CuAm)/d$$

(en inversant p et q, n et m)

-Calcul de D:

$$\begin{aligned} D &= [(EuCm - CuEm)(AuCn - CuAn) - (EuCn - CuEn)(AuCm - CuAm)]/d \\ &= [EuCmAuCn - EuCmCuAn - CuEmAuCn + CuEmCuAn \\ &\quad - (EuCnAuCm - EuCnCuAm - CuEnAuCm + CuEnCuAm)]/d \\ &= [-EuCmAn - EmAuCn + EmCuAn + EuCnAm + EnAuCm - EnCuAm]Cu/d \\ &= [Au(EnCm - EmCn) + Eu(CnAm - CmAn) + Cu(EmAn - EnAm)]Cu/d \\ &= K Cu/d \end{aligned}$$

D'où on en déduit, puisque  $Pi = N/D$ :

$$Pi = \frac{Au}{Cu} = \frac{a11U + a12V + a13W}{a31U + a32V + a33W}$$

Vus les rôles symétriques de  $P_i$  et de  $P_j$ , on en déduit:

$$P_j = \frac{E_u}{C_u} = \frac{a_{21}U + a_{22}V + a_{23}W}{a_{31}U + a_{32}V + a_{33}W}$$

Ceci démontrant en même temps que toutes les droites images de droites parallèles se coupent en un même point, et les formules (15).

Les formules (16) s'obtiennent simplement en remplaçant le vecteur  $\vec{u}$  successivement par  $\vec{X}$ ,  $\vec{Y}$  et  $\vec{Z}$ .

Montrons maintenant les formules (18). Pour cela, on fait les hypothèses (17):

$$\begin{cases} \vec{P}_f = (P_i, P_j, 1) \\ \vec{C}_1 = (a_{11}, a_{21}, a_{31}) \\ \vec{C}_2 = (a_{12}, a_{22}, a_{32}) \\ \vec{C}_3 = (a_{13}, a_{23}, a_{33}) \end{cases}$$

Commençons par calculer les produits vectoriels  $(\vec{C}_3 \wedge \vec{C}_2)$ ,  $(\vec{C}_1 \wedge \vec{C}_3)$  et  $(\vec{C}_2 \wedge \vec{C}_1)$ .

-Calcul de  $(\vec{C}_3 \wedge \vec{C}_2)$ ,  $(\vec{C}_1 \wedge \vec{C}_3)$  et  $(\vec{C}_2 \wedge \vec{C}_1)$ .

$$(32) \quad \begin{aligned} (\vec{C}_3 \wedge \vec{C}_2) &= (a_{23}a_{32} - a_{22}a_{33}, a_{12}a_{33} - a_{13}a_{32}, a_{13}a_{22} - a_{12}a_{23}) \\ (\vec{C}_1 \wedge \vec{C}_3) &= (a_{21}a_{33} - a_{23}a_{31}, a_{13}a_{31} - a_{11}a_{33}, a_{11}a_{23} - a_{13}a_{21}) \\ (\vec{C}_2 \wedge \vec{C}_1) &= (a_{22}a_{31} - a_{21}a_{32}, a_{11}a_{32} - a_{12}a_{31}, a_{12}a_{21} - a_{11}a_{22}) \end{aligned}$$

En utilisant (15), calculons  $U, V$  en fonction de  $P_i, P_j, W$ . Si on multiplie par les dénominateurs, on obtient:

$$(33) \quad \begin{aligned} P_i(a_{31}U + a_{32}V + a_{33}W) &= a_{11}U + a_{12}V + a_{13}W \\ P_j(a_{31}U + a_{32}V + a_{33}W) &= a_{21}U + a_{22}V + a_{23}W \\ U(P_i a_{31} - a_{11}) + V(P_i a_{32} - a_{12}) &= -W(P_i a_{33} - a_{13}) \\ U(P_j a_{31} - a_{21}) + V(P_j a_{32} - a_{22}) &= -W(P_j a_{33} - a_{23}) \end{aligned}$$

On élimine  $V$  dans (33):

$$\begin{aligned} U[(P_i a_{31} - a_{11})(P_j a_{32} - a_{22}) - (P_j a_{31} - a_{21})(P_i a_{32} - a_{12})] &= \\ W[-(P_i a_{33} - a_{13})(P_j a_{32} - a_{22}) + (P_j a_{33} - a_{23})(P_i a_{32} - a_{12})] & \\ U[\cancel{P_i a_{31}} P_j a_{32} - P_i a_{31} a_{22} - a_{11} P_j a_{32} + a_{11} a_{22} - & \\ (P_j a_{31} P_i a_{32} - P_j a_{31} a_{12} - a_{21} P_i a_{32} + a_{21} a_{12})] &= \\ W[-(P_i a_{33} P_j a_{32} - P_i a_{33} a_{22} - a_{13} P_j a_{32} + a_{13} a_{22}) + & \\ P_j a_{33} P_i a_{32} - P_j a_{33} a_{12} - a_{23} P_i a_{32} + a_{23} a_{12}] & \end{aligned}$$

On remarque que les termes en  $ij$  s'annulent, on obtient:

$$(34) \quad \begin{aligned} U[P_i(a_{21}a_{32} - a_{31}a_{22}) + P_j(a_{31}a_{12} - a_{11}a_{32}) + 1(a_{11}a_{22} - a_{21}a_{12})] &= \\ W[P_i(a_{33}a_{22} - a_{23}a_{32}) + P_j(a_{13}a_{32} - a_{33}a_{12}) + 1(a_{23}a_{12} - a_{13}a_{22})] & \\ U[-(\vec{C}_2 \wedge \vec{C}_1) \cdot \vec{P}_f] &= W[-(\vec{C}_3 \wedge \vec{C}_2) \cdot \vec{P}_f] \\ U &= W[(\vec{C}_3 \wedge \vec{C}_2) \cdot \vec{P}_f] / [(\vec{C}_2 \wedge \vec{C}_1) \cdot \vec{P}_f] \end{aligned}$$

On élimine U dans (33):

$$V[(Pia32-a12)(Pja31-a21)-(Pja32-a22)(Pia31-a11)] = \\ W[-(Pia33-a13)(Pja31-a21)+(Pja33-a23)(Pia31-a11)]$$

$$V[\cancel{Pia32}Pja31-\cancel{Pia32}a21-a12Pja31+a12a21- \\ (Pja32\cancel{Pia31}-Pja32a11-a22Pia31+a22a11)] = \\ W[-(\cancel{Pia33}Pja31-\cancel{Pia33}a21-a13Pja31+a13a21)+ \\ Pja33\cancel{Pia31}-Pja33a11-a23Pia31+a23a11]$$

On remarque que les termes en ij s'annulent, on obtient:

$$V[Pi(a22a31-a32a21)+Pj(a32a11-a12a31)+l(a12a21-a22a11)] = \\ W[Pi(a33a21-a23a31)+Pj(a13a31-a33a11)+l(a23a11-a13a21)]$$

$$V[(\vec{C2} \wedge \vec{C1}) \cdot \vec{Pf}] = W[(\vec{C1} \wedge \vec{C3}) \cdot \vec{Pf}]$$

$$(35) \quad V = W[(\vec{C1} \wedge \vec{C3}) \cdot \vec{Pf}] / [(\vec{C2} \wedge \vec{C1}) \cdot \vec{Pf}]$$

On sait que le vecteur  $\vec{u}:(U,V,W)$  représente une direction, par conséquent une de ses variables est libre. Si on pose  $W=(\vec{C2} \wedge \vec{C1}) \cdot \vec{Pf}$ , on obtient les formules (18) avec (34) et (35).

#### 7.4 Démonstration 4

Il s'agit de démontrer la formule (24) du paragraphe 7.1:

$$(A \wedge B) \wedge (A \wedge C) = [A \cdot (B \wedge C)] \cdot A$$

Calculons les termes  $(A \wedge B)$ ,  $(A \wedge C)$  et  $(B \wedge C)$ . On pose:

$$\begin{cases} A=(a1, a2, a3) \\ B=(b1, b2, b3) \\ C=(c1, c2, c3) \end{cases}$$

On a:

$$\begin{aligned} (A \wedge B) &= (a2b3-b2a3, b1a3-a1b3, a1b2-b1a2) \\ (A \wedge C) &= (a2c3-c2a3, c1a3-a1c3, a1c2-c1a2) \\ (B \wedge C) &= (b2c3-c2b3, c1b3-b1c3, b1c2-c1b2) \end{aligned}$$

Calculons le terme  $(A \wedge B) \wedge (A \wedge C)$ .

$$(A \wedge B) \wedge (A \wedge C) = ( \\ (b1a3-a1b3)(a1c2-c1a2) - (c1a3-a1c3)(a1b2-b1a2), \\ (a2c3-c2a3)(a1b2-b1a2) - (a2b3-b2a3)(a1c2-c1a2), \\ (a2b3-b2a3)(c1a3-a1c3) - (a2c3-c2a3)(b1a3-a1b3) \\ )$$

Calculons le premier élément de ce vecteur, il est égal à:

$$\begin{aligned} b1a3a1c2 - b1a3c1a2 - a1b3a1c2 + a1b3c1a2 - \\ (c1a3a1b2 - c1a3b1a2 - a1c3a1b2 + a1c3b1a2) = \\ a1(b1a3c2 - b3a1c2 + b3c1a2 - c1a3b2 + c3a1b2 - c3b1a2) = \\ a1[a1(c3b2 - b3c2) + a2(b3c1 - c3b1) + a3(b1c2 - c1b2)] = \\ a1[A \cdot (B \wedge C)] \end{aligned}$$

Calculons le second élément de ce vecteur, il est égal à :

$$\begin{aligned} & a2c3a1b2 - a2c3b1a2 - c2a3a1b2 + c2a3b1a2 - \\ & \quad (a2b3a1c2 - a2b3c1a2 - b2a3a1c2 + b2a3c1a2) = \\ & a2(c3a1b2 - c3b1a2 + c2a3b1 - b3a1c2 + b3c1a2 - b2a3c1) = \\ & a2[a1(c3b2 - b3c2) + a2(b3c1 - c3b1) + a3(c2b1 - b2c1)] = \\ & a2[A \cdot (B \wedge C)] \end{aligned}$$

Calculons le troisième élément de ce vecteur, il est égal à :

$$\begin{aligned} & a2b3c1a3 - a2b3a1c3 - b2a3c1a3 + b2a3a1c3 - \\ & \quad (a2c3b1a3 - a2c3a1b3 - c2a3b1a3 + c2a3a1b3) = \\ & a3(a2b3c1 - b2c1a3 + b2a1c3 - a2c3b1 + c2b1a3 - c2a1b3) = \\ & a3[a1(b2c3 - c2b3) + a2(b3c1 - c3b1) + a3(c2b1 - b2c1)] = \\ & a3[A \cdot (B \wedge C)] \end{aligned}$$

On a donc :

$$\begin{aligned} (A \wedge B) \wedge (A \wedge C) &= (a1[A \cdot (B \wedge C)], a2[A \cdot (B \wedge C)], a3[A \cdot (B \wedge C)]) \\ &= [A \cdot (B \wedge C)] \cdot (a1, a2, a3) \\ &= [A \cdot (B \wedge C)] \cdot A \end{aligned}$$

Ce qu'il fallait démontrer.

## ANNEXE E

## CALCUL OPTIMISE DU MEILLEUR PLAN

## EN UN POINT D'UNE IMAGE 3D

Il s'agit de calculer le meilleur plan au sens des moindres carrés, passant par 9 points (cf. chapitre 4, paragraphe 2.1.1). Ces 9 points sont répartis dans l'espace de la manière suivante ( par exemple, le point "Y2" est situé en (X,Z+tZ,Y2)):

Z+tZ	: Y1	: Y2	: Y3	:
	:.....	:.....	:.....	:
Z	: Y4	: Y5	: Y6	:
	:.....	:.....	:.....	:
Z-tZ	: Y7	: Y8	: Y9	:
	:.....	:.....	:.....	:
	X-tX	X	X+tX	

Une optimisation est possible parce que les 9 points sont disposés régulièrement. On a:

$$X1=X4=X7=X-tX, \quad X2=X5=X8=X, \quad X3=X6=X9=X+tX$$

$$Z1=Z2=Z3=Z+tZ, \quad Z4=Z5=Z6=Z, \quad Z7=Z8=Z9=Z-tZ$$

L'équation du plan à calculer est de la forme  $Y=aX+bZ+c$ . Cette équation est en rapport avec le type d'image 3D fourni par le capteur qui est  $Y=f(X,Z)$ . Il faut calculer la valeur:

$$Q = \min_{a,b,c} \sum_{i=1}^{i=9} (Y_i - (aX_i + bZ_i + c))^2$$

$$\text{Si l'on pose} \quad S = \sum_{i=1}^{i=9} (Y_i - (aX_i + bZ_i + c))^2$$

alors, il faut qu'on ait les dérivées de S par rapport aux variables a, b et c nulles, c'est-à-dire:

$$\frac{dS}{da} = \frac{dS}{db} = \frac{dS}{dc} = 0$$

Cela conduit à un système linéaire symétrique (1) de 3 équations à 3 inconnues qui est le suivant (après dérivation):

$$(1) \begin{cases} \sum X_i^2 a + \sum X_i Z_i b + \sum X_i c = \sum X_i Y_i \\ \sum X_i Z_i a + \sum Z_i^2 b + \sum Z_i c = \sum Z_i Y_i \\ \sum X_i a + \sum Z_i b + \sum 1 c = \sum Y_i \end{cases}$$

Pour plus de commodité, on notera:

$$\begin{cases} SY = \sum_{i=1}^{i=9} Y_i \\ SY2 = \sum_{i=1}^{i=9} Y_i^2 \\ GX = Y_3 + Y_6 + Y_9 - Y_1 - Y_4 - Y_7 \\ GZ = Y_1 + Y_2 + Y_3 - Y_7 - Y_8 - Y_9 \end{cases}$$

On calcule les différents coefficients, le système (1) donne:

$$\begin{aligned} (2) & \begin{cases} (9X^2 + 6tX^2)a + (9XZ) b + 9X c = X SY + tX GX \\ (9XZ) a + (9Z^2 + 6tZ^2) b + 9Z c = Z SY + tZ GZ \\ 9X a + 9Z b + 9 c = SY \end{cases} \\ (3) & \\ (4) & \end{aligned}$$

En exemple, nous donnons le calcul du coefficient  $SX2 = \sum_{i=1}^{i=9} X_i^2$ .

$$\begin{aligned} SX2 &= 3(X-tX)^2 + 3X^2 + 3(X+tX)^2 = 3(X^2 - 2XtX + tX^2) + 3X^2 + 3(X^2 + 2XtX + tX^2) \\ &= 9X^2 + 6tX^2 \end{aligned}$$

En additionnant l'équation (2) avec l'équation (4), multipliée par  $-X$ , on a:

$$(9X^2 + 6tX^2 - 9X^2)a + (9XZ - 9ZX)b + (9X - 9X)c = X SY + tX GX - X SY$$

ce qui donne  $6tX^2 a = tX GX$ , soit  $6 tX a = GX$ , car  $tX \neq 0$ . On a donc:

$$(5) \quad a = \frac{GX}{6 tX}$$

On obtient de même, pour b:

$$(6) \quad b = \frac{GZ}{6 tZ}$$

L'équation (4) nous donne, pour c:

$$(7) \quad c = \frac{SY}{9} - a X - b Z$$

Il est nécessaire, maintenant, de calculer l'écart quadratique Q, égal à S, quand on remplace a, b et c par les valeurs données par (5), (6) et (7).

$$Q = (Y1 - (a(X-tX)+b(Z+tZ)+c))^2 + (Y2 - (a X + b(Z+tZ)+c))^2 + \\ (Y3 - (a(X+tX)+b(Z+tZ)+c))^2 + (Y4 - (a(X-tX)+b Z + c))^2 + \\ (Y5 - (a X + b Z + c))^2 + (Y6 - (a(X+tX)+b Z + c))^2 + \\ (Y7 - (a(X-tX)+b(Z-tZ)+c))^2 + (Y8 - (a X + b(Z-tZ)+c))^2 + \\ (Y9 - (a(X+tX)+b(Z-tZ)+c))^2$$

$$Q = SY^2 - 2( \\ Y1 (a(X-tX)+b(Z+tZ)+c) + Y2 (a X + b(Z+tZ)+c) + \\ Y3 (a(X+tX)+b(Z+tZ)+c) + Y4 (a(X-tX)+b Z + c) + \\ Y5 (a X + b Z + c) + Y6 (a(X+tX)+b Z + c) + \\ Y7 (a(X-tX)+b(Z-tZ)+c) + Y8 (a X + b(Z-tZ)+c) + \\ Y9 (a(X+tX)+b(Z-tZ)+c) \\ ) + \\ (a(X-tX)+b(Z+tZ)+c)^2 + (a X + b(Z+tZ)+c)^2 + \\ (a(X+tX)+b(Z+tZ)+c)^2 + (a(X-tX)+b Z + c)^2 + \\ (a X + b Z + c)^2 + (a(X+tX)+b Z + c)^2 + \\ (a(X-tX)+b(Z-tZ)+c)^2 + (a X + b(Z-tZ)+c)^2 + \\ (a(X+tX)+b(Z-tZ)+c)^2$$

$$\begin{aligned}
Q = & SY^2 - 2 \left( \frac{aX+bZ+c}{2} SY + a \frac{tXGX}{2} + b \frac{tZGZ}{2} \right) + \\
& a \left( \frac{3(X-tX)^2}{2} + 3X^2 + \frac{3(X+tX)^2}{2} \right) + \\
& b \left( \frac{3(Z+tZ)^2}{2} + 3Z^2 + \frac{3(Z-tZ)^2}{2} \right) + 9c + \\
& 2ab \left( \frac{(X-tX)(Z+tZ)+X(Z+tZ)+(X+tX)(Z+tZ)}{2} + \right. \\
& \quad \left. \frac{(X-tX)Z}{2} + \frac{XZ}{2} + \frac{(X+tX)Z}{2} + \right. \\
& \quad \left. \frac{(X-tX)(Z-tZ)+X(Z-tZ)+(X+tX)(Z-tZ)}{2} \right) + \\
& 2ac \left( \frac{(X-tX)+X+(X+tX)}{2} + \right. \\
& \quad \left. \frac{(X-tX)+X+(X+tX)}{2} + \right. \\
& \quad \left. \frac{(X-tX)+X+(X+tX)}{2} \right) + \\
& 2bc \left( \frac{(Z+tZ)+(Z+tZ)+(Z+tZ)}{2} + \right. \\
& \quad \left. \frac{Z+Z+Z}{2} + \right. \\
& \quad \left. \frac{(Z-tZ)+(Z-tZ)+(Z-tZ)}{2} \right)
\end{aligned}$$

On sait par (4) que  $aX+bZ+c=SY/9$ . De plus (5) et (6) nous donnent:

$$atXGX = atX \frac{6atX}{2} = 6a \frac{tX^2}{2}$$

$$btZGZ = btZ \frac{6btZ}{2} = 6b \frac{tZ^2}{2}$$

En utilisant ces expressions et en développant la formule de Q ci-dessus, on obtient:

$$\begin{aligned}
Q = & SY^2 - 2 \left( \frac{SY^2}{9} + 6a \frac{tX^2}{2} + 6b \frac{tZ^2}{2} \right) + \\
& a \left( \frac{9X^2}{2} + 6tX^2 \right) + b \left( \frac{9Z^2}{2} + 6tZ^2 \right) + 9c + \\
& 2ab \left( \frac{(X-tX)3Z+X3Z+(X+tX)3Z}{2} \right) + \\
& 2ac \left( \frac{3X+3X+3X}{2} \right) + 2bc \left( \frac{3Z+3Z+3Z}{2} \right) \\
= & SY^2 - \frac{2}{9} SY^2 - 12a \frac{tX^2}{2} - 12b \frac{tZ^2}{2} + \\
& 9a \frac{X^2}{2} + 6a \frac{tX^2}{2} + 9b \frac{Z^2}{2} + 6b \frac{tZ^2}{2} + \\
& 2ab(9XZ) + 2ac(9X) + 2bc(9Z) + 9c \\
= & SY^2 - \frac{2}{9} SY^2 - 6a \frac{tX^2}{2} - 6b \frac{tZ^2}{2} + \\
& 9a \frac{X^2}{2} + 9b \frac{Z^2}{2} + 18abXZ + 18acX + 18bcZ + 9c \\
= & SY^2 - \frac{2}{9} SY^2 - 6a \frac{tX^2}{2} - 6b \frac{tZ^2}{2} + 9(aX+bZ+c) \\
= & SY^2 - \frac{2}{9} SY^2 - 6a \frac{tX^2}{2} - 6b \frac{tZ^2}{2} + 9(SY/9) \\
= & SY^2 - \frac{2}{9} SY^2 - 6a \frac{tX^2}{2} - 6b \frac{tZ^2}{2} + SY/9
\end{aligned}$$

$$= SY^2 - SY^2/9 - 6a tX - 6b tZ$$

Vu les valeurs de a et de b données par (5) et (6), on obtient finalement:

$$(8) \quad Q = SY^2 - \frac{SY^2}{9} - \frac{GX^2}{6} - \frac{GZ^2}{6}$$

Il vaut mieux prendre  $|Q|$  car, bien que Q soit toujours théoriquement positif ou nul (somme de carrés), quand sa valeur est proche de zéro, les erreurs d'arrondi peuvent rendre sa valeur négative.

Si on s'intéresse à un écart moyen, qui correspondrait à la moyenne des écarts des points au plan, il faut prendre la valeur:

$$(9) \quad E = \frac{|Q|^{1/2}}{9}$$

On notera la simplicité du résultat, après calcul. Si on évalue le nombre d'opérations, on a (9"\*,8"+) pour SY<sup>2</sup>, (8"+) pour SY, (5"+) pour GX, (5"+) pour GZ, (1"\*\*,1"/") pour SY\*SY/9, (1"\*\*,1"/") pour GX\*GX/6, (1"\*\*,1"/") pour GZ\*GZ/6, (3"+) pour Q, et (1"/") et une racine carrée pour E. Cela fait 12 multiplications, 29 additions ou soustractions, 4 divisions et une racine carrée. La racine carrée peut être calculée avec une table et un procédé d'interpolation, dans ce cas cela se ramène à un accès mémoire plus quelques multiplications.



A U T O R I S A T I O N D E S O U T E N A N C E

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU le rapport de présentation de Monsieur le Professeur J.C LATOMBE

**Monsieur Paul-Louis BORIANNE**

est autorisé à présenter une thèse en soutenance pour l'obtention du titre de DOCTEUR  
de TROISIEME CYCLE, spécialité "Informatique".

Fait à Grenoble, le 20 mars 1984

Le Président de l'I.N.P.-G

**D. BLOCH**

Président

de l'Institut National Polytechnique  
de Grenoble

*P.O. le Vice-Président,*



Thèse de 3ème cycle

Auteur: Paul-Louis BORIANNE

Etablissement: LIFIA/IMAG

Titre: CONTRIBUTIONS A LA VISION PAR ORDINATEUR  
TRIDIMENSIONNELLE.

Résumé:

Notre travail est organisé autour du thème de la vision tridimensionnelle. Le problème de la modélisation automatique d'objets est abordé, ainsi que celui de la collaboration d'un capteur 3D et d'un capteur 2D. Cette modélisation est effectuée en trois temps: - acquisition d'une image 3D grâce à un capteur réalisé au cours de ce travail - obtention de modèles partiels grâce à un procédé de segmentation d'images 3D - obtention de modèles complets grâce à une unité de génération d'hypothèses de correspondances 3D et à une unité de fusion de modèles partiels.

Mots clefs: Capteur 3D, Vision 3D, Modélisation automatique, Segmentation en régions, Correspondance d'images.